



TAMPEREEN TEKNILLINEN YLIOPISTO

JUHA NURMI
DATALÄHTÖINEN VIRTUAALINEN KONELABORATORIO
MEKATRONISEN KONEEN YHTEISSUUNNITTELUN
TUKENA

Diplomityö

Tarkastajat:
dosentti Ossi Nykänen
projektipäällikkö Pekka Ranta
Tarkastajat ja aihe hyväksytyt
Luonnontieteiden ja
ympäristötekniikan tiedekunnan
kokouksessa 03.10.2012

TIIVISTELMÄ

TAMPEREEN TEKNILLINEN YLIOPISTO

Teknis-luonnontieteellinen koulutusohjelma

NURMI, JUHA: Datalähtöinen virtuaalinen konelaboratorio mekatronisen koneen yhteissuunnittelun tukena

Diplomityö, 73 sivua, 13 liitesivua

Huhtikuu 2013

Pääaine: Hypermedia

Tarkastajat: dosentti Ossi Nykänen ja projektipäällikkö Pekka Ranta

Avainsanat: ohjelmistoarkkitehtuurit, semanttinen tiedon mallinnus, RDF, OWL, mekatroniikka, yhteissuunnittelun tuki, web-ohjelmointi, datalähtöisyys, virtuaalilaboratorio

Mekatronisen koneen suunnittelu vaatii useita teknologioita yhdistävää suunnittelutyötä. Yhteissuunnittelu toisi suunnittelijat yhteen suunnittelemaan kokonaista toimivaa laitetta yksittäisten suunnitteluteknologioiden toteuttamisen sijaan. Kuitenkin suunnittelutiedon automatisoitu yhdisteleminen on hankalaa, koska suunnitteluohjelmat eivät tue yhteissuunnittelua ja suunnittelutietojen yhdistelyä riittävästi. Koska yhteissuunnittelu vaatii yhteisen suunnittelutyöpöydän, jossa suunnittelutieto yhdistyy yhdeksi kokonaisuudeksi, on mielekästä tutkia tällaista konseptia sekä sen toteuttamista.

Virtuaalinen konelaboratorio (VML) tarjoaa yhteisen näkymän mekatroniseen laitteeseen. Diplomityössä selvitetään, millainen on hyvä suunnittelua tukeva VML-järjestelmä, ja miten VML-järjestelmä kannattaa toteuttaa.

Työ jakautuu kolmeen osaan: Kirjallisuusselvitykseen virtuaalikonelaboratorios- ta, suunnittelusta sekä soveltuvista toteutusteknologioista VML-järjestelmän toteuttamiseksi; asiantuntijahaastatteluun, jossa kerätään ideoita ja palautetta siihen, millaisia ongelmia yhteinen virtuaalinen suunnittelutyöpöytä voisi ratkaista; ja tutkimusprototyypin toteuttamiseen, jossa selvitetään teknisiä mahdollisuuksia toteuttaa automatisoidusti tällainen toimiva VML-järjestelmä suunnitteluohjelmien tuot- tamasta tiedosta.

Prototyypin toteuttamisessa käytetään valmiin jo toteutetun toimivan puomi- laitteen suunnitteluaineistoa. Valmiista prototyypistä arvioidaan käytettyjen webin teknologioiden sopivuutta tällaisen yhteissuunnittelun tukityövälineen toteuttami- seksi.

Tutkimus osoittaa, että yhteissuunnittelulle on selkeä tarve mekatronisen koneen suunnittelussa, mutta nykyinen suunnitteluprosessi ja -välineet eivät tue tällaista suunnittelua riittävästi. Havaitaan myös, että tietokoneavusteisen tuen toteuttami- nen vaatii suunnittelutiedon yhdistelyä aivan uudella tavalla. VML täytyy olla to- teutettavissa datalähtöisesti suunnittelutiedosta, jotta suunnittelijat voivat seurata koko ajan, miten heidän suunnittelutyönsä edistää yhteisiä tavoitteita.

ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY

Master's Degree Programme in Science and Engineering

NURMI, JUHA: Data-driven virtual views to support mechatronics machine design process

Master of Science Thesis, 73 pages, 13 Appendix pages

April 2013

Major: Hypermedia

Examiners: Adjunct Professor Ossi Nykänen and Project Manager Pekka Ranta

Keywords: software architectures, semantic knowledge modeling, RDF, OWL, mechatronics, co-design, web development, data-driven programming, virtual laboratory

Designing mechatronic systems requires co-operation between various design disciplines. To help this task co-design should result a process of creating one working machine, not only focusing on designing working subsystems. Unfortunately, it is not automatically possible to merge design information together because CAD tools does neither support co-design nor sufficiently combine data. A shared virtual design space might be useful to understanding combined design data. As such, it is worthwhile to study this kind of computer aided design concept and how it could be developed.

Virtual Machine Laboratory (VML) provides shared view to a mechatronic machine. This thesis investigates the questions of what sort of VML software is able to support the design tasks, and what kind of software development is needed to generate this kind of VML.

This thesis consists of three main parts: literature study, interview research and prototype software implementation. In the literature study part, VML systems, design process and suitable technologies to implement VML are explored. In the interview research part, experts are giving their views about VML support to a machine design process. In the third part, a prototype software is implemented to explore how VML can be generated from the data produced by various design tools.

The design data of an already built boom is a subject for a case study. Prototype VML uses this data and with the generated views to the data we are able to evaluate how semantic web technologies are suitable for producing computer-supported cooperative work tools for co-design.

The study indicates that co-design is clearly needed for machine design, but the process and the tools to support co-design are insufficient. The result of this study suggests that we need to model and use the knowledge itself in new ways. VML needs data-driven implementation from the design data produced by designers with their CAD tools. Like this, the designers are able to monitor how their cooperative work helps the whole design process forward.

ALKUSANAT

Tämä diplomityö on tehty Tampereen teknillisen yliopiston matematiikan laitoksen Intelligent Information Systems Laboratoriossa Smart Simulators -tutkimusryhmässä.

Työn tarkastajat ovat dosentti Ossi Nykänen ja projektipäällikkö Pekka Ranta. Heille kuuluu suuri kiitos työn asiantuntevasta ohjaamisesta.

Loistavasta työilmapiiristä tulee tietysti kiittää kaikkia projektin työtovereita. Erityisesti tutkija Jaakko Salonen, tutkimusapulainen Matti Helminen ja tutkija Markus Rokala olivat läheisesti mukana tekemässä tähänkin diplomityöhön liittyvää työtä ja sain heiltä apua sekä ideoita.

Semogen-hankkeen yrityskumppanit - jotka ovat AEL, Cargotec Oyj, Etteplan Design Center Oy, Sandvik Mining and Construction Oy ja Vertex Systems Oy - tarjosivat merkittävää panosta diplomityön avuksi.

Erityistä apua sain Callimachus-ohjelmistoa kehittävältä 3 Round Stones -yritykseltä; heidän teknologiajohtajansa tohtori David Wood, joka on myös mukana aktiivisesti W3C:ssä kehittämässä semanttisen webin arkkitehtuuria, vastaili ystävällisesti IRC:n välityksellä kaikkiin mahdollisiin kysymyksiin Callimachus-ohjelmistosta.

Ystävät ovat olleet myös osaltaan avuksi diplomityössäni. Kalle raahasi minua kahville ja syömään diplomityöni äärestä aina työpäivieni aikana. Juuso vastaili asiantuntevasti kysymyksiini koneenrakentamisesta. Teemun kanssa kävimme kuntosalilla.

Avovaimoni Heidi piti minusta huolta diplomityön aikana. Hän on rinnallani muistuttamassa minulle, mikä on elämässä tärkeää ja kuinka vaikeinkin ongelma on lopulta ratkaistavissa yhteistyöllä ja rakkaudella ♡

Tampereella 2. huhtikuuta 2013

Juha Nurmi

SISÄLLYS

1. Johdanto	1
2. Virtuaalikonelaboratoriot	4
2.1 VML:n käyttömahdollisuudet	9
2.2 VML:n toiminnot	9
3. Näkökulmia suunnitteluun	12
3.1 Tietokoneavusteinen yhteisöllinen työ	12
3.2 Mekatronisen koneen suunnittelu	14
3.2.1 Suunnitteluprosessi	15
3.2.2 Suunnittelutyövälineet	16
3.2.3 Suunnittelutiedon tallennusmuodot	17
3.3 Suunnittelumalleja	19
3.4 Yhteissuunnittelu	19
4. Semanttisen webin sovellustekniikat	24
4.1 Semanttinen tiedon mallinnus	24
4.1.1 RDF	25
4.1.2 Skeemat ja ontologiat	27
4.1.3 RDFa	33
4.2 Web-pohjaiset sovellukset	34
4.2.1 Käyttöliittymä selaimessa	34
4.3 Datalähtöisyys ohjelmistokehityksessä	36
4.3.1 Datalähtöisyys paradigmana	36
4.3.2 Semanttisen tietomallin hallinta ja jakelu	36
4.3.3 Webin teknologioita hyödyntäviä esimerkkijärjestelmiä	36
5. Tapaustutkimus asiantuntijahaastattelulla	39
5.1 VML:n ominaisuuksien kartoitus	39
5.2 Kyselyn tulokset	40
5.2.1 Johtopäätökset kyselyn tuloksista	44
6. Toteutettu tutkimusprototyyppi	49
6.1 Toteutustapa	50
6.2 Toteutetut ominaisuudet	51
6.2.1 Vaatimusten ja toimintomallin hallinta	56
6.3 Generoinnin vaatimukset	60
6.3.1 Putkilinjakuvaus	61
6.3.2 Tietomallin vaatimukset	61
6.3.3 Vaatimukset suunnitteluohjelmistoille	62
7. Arviointi ja yhteenveto	63
7.1 Vertailu muihin virtuaalilaboratorioihin	63

7.2 Yhteenveto	66
Lähteet	69
A.Liitteitä	i
A.1 Rinnakkaistettu simulaation kuuntelu WebSocket-yhteydellä	ii
A.2 Toimintomallin semanttinen kuvaus	vi
A.3 Semanttinen datalehti	xii

LYHENTEET JA TERMIT

CAD	<i>Computer-aided Design.</i> Tietokoneavusteinen suunnittelu.
Callimachus	Callimachus on sovelluskehys datalähtöisten semanttisen webin sovellusten rakentamiseen.
CAN	<i>Controller Area Network.</i> Koneiden väylästandardi, jolla yhdistetään kommunikoi- vat osat keskenään.
CSCW	<i>Computer-supported cooperative work.</i> Tietokoneavusteinen yhteisöllinen työ. Tarkoittaa ihmisten ryhmätyöskentelyn avustamista tie- totekniiikan avulla.
Datalehti	<i>Datasheet.</i> Dokumentti, joka kuvaa komponentin tai koneen tekni- set ominaisuudet.
DITA	<i>Darwin Information Typing Architecture.</i> XML-pohjainen arkkitehtuuri teknisen dokumentaation tuottamiseen, hallintaan ja julkaisemiseen.
DOM	<i>Document Object Model.</i> Selaimen toteuttama rajapinta, jota käytetään sisällön käsittelyyn JavaScript-ohjelmointikielellä.
HTML5	<i>Hypertext Markup Language</i> -standardin viides versio.
HTTP	<i>Hypertext Transfer Protocol.</i> Sovelluserroksen tiedonsiirtoprotokolla selaimen ja HTTP-palvelimen välillä.
IISLab	<i>Intelligent Information Systems Laboratory.</i> Tampereen teknillisen yliopiston matematiikan laitok- sen yksikkö.
JavaScript	Selaimen tukema ohjelmointikieli.

Kvantifointi	Lukumääräisten suureiden tuottaminen laadullisesta sanallisesta aineistosta.
MCDM	<i>Multiple-criteria decision-making.</i> Vertailumenetelmä, joka perustuu järjestelmän kriteerien ja niiden tärkeyden määrittelemiseen.
Mekatroniikka	Suunnitteluosa-alueet yhdistävä monitekninen tieteenala, jonka avulla tuotetaan konejärjestelmiä.
Ontologia	Semanttisen webin ontologia on menetelmä käsitteiden ja käsitteiden välisten suhteiden kuvaamiseen.
OWL	<i>Web Ontology Language.</i> Ontologiatiedon kuvailukieli koneellista päättelyä varten.
PDM	<i>Product Data Management.</i> Tuotetiedon hallintajärjestelmä.
RDF	<i>Resource Description Framework.</i> Tiedon kuvailukielen standardi.
RDFa	<i>Resource Description Framework in Attributes.</i> Standardi rakenteisen metadatan lisäämisestä XHTML-dokumentteihin.
RDFS	<i>RDF Schema.</i> Yksinkertainen skeeman kuvailukieli koneellista päättelyä varten.
Selain	WWW-selainohjelma. Käytännössä esimerkiksi Google Chrome 19 tai Firefox 13.
Semantiikka	Semanttisen webin näkökulmasta tiedon merkityksen kuvailu koneluettavaksi.
SPARQL	<i>SPARQL Protocol and RDF Query Language.</i> RDF:ksi mallinnetulle tiedolle tarkoitettu kyselykieli.

SVG	<i>Scalable Vector Graphics.</i> XML-pohjainen kuvauskieli 2D-grafiikalle.
TTY	Tampereen teknillinen yliopisto
VL	<i>Virtual laboratory.</i> Virtuaalinen laboratorio. Aidon tuntuinen virtuaalinen testiympäristö.
VML	<i>Virtual machine laboratory.</i> Virtuaalinen konelaboratorio.
W3C	<i>World Wide Web Consortium.</i>
WebGL	<i>Web-based Graphics Language.</i> Mahdollistaa suorituskykyisen 3D-grafiikan selaimessa.
Wille	<i>Wille Visualisation System.</i> Sovelluskehys tiedon visualisoimiseksi.
X3D	XML-pohjainen kuvauskieli 3D-grafiikalle. X3DOM:n käyttämä kuvauskieli.
X3DOM	JavaScript-kirjasto WebGL-piirtoon.
XML	<i>Extensible Markup Language.</i> Merkintäkieli rakenteisille dokumenteille.
XSLT	<i>Extensible Stylesheet Language Transformations.</i> XML-dokumenttien muunnosprosessointikieli.

1. JOHDANTO

Mekatroniikka on useita teknologioita yhdistävä monitekninen suunnitteluala [10]. Mekatronisten järjestelmien - kuten esimerkiksi metsäkoneet tai lentokoneet - suunnittelu vaatii useiden erilaisten suunnittelualojen yhteistyötä. Kuitenkin suunnittelun aikana on ongelmallista tarkastella esimerkiksi mekaniikka- ja sähkösuunnittelun tuottaman suunnittelun yhteistulosta. Tämän seurauksena saattaa suunnittelussa syntyä yhteensopimattomia teknologiaratkaisuja, jotka ovat sitä kalliimpia korjata, mitä myöhemmin ne havaitaan [28]. Jotta suunnittelun rajapinnat eri teknologioiden välillä olisivat aina yhteensopivia ja toteuttaisivat yhdessä järjestelmään suunniteltuja toimintoja, tarvitaan menetelmiä, malleja ja automatisointia suunnittelun tuottaman toiminnallisuuden toteutumisen tarkasteluun [36].

Diplomityössä tarkastellaan mekatronisen konejärjestelmän suunnittelun tukemista prototypoiduilla yhteissuunnittelun tukityövälineillä. Tutkimusryhmän visio on tuoda suunnitteluinformaatio yhteen ja tarkastella sitä yhteisessä suunnittelutilassa. Visiona on luoda näkymä ja työväline suunnitteluun sitä varten, että kaikki suunnitteluprojektin jäsenet voisivat yhdessä tarkastella suunnitteluratkaisujensa toimivuutta koko suunnittelutyön ajan. Näin voidaan tarjota mahdollisuudet parempaan yhteissuunnitteluun, jossa kaikki yhdistävät työtään löytääkseen ratkaisut suunnitteluvaatimuksiin.

Eräs lähestymistapa mekatroniseen yhteissuunnitteluun on luoda tietokoneavusteisesti yhteinen virtuaalinäkymä suunnittelutietoon sekä yhteistoiminnalliset suunnittelutyövälineet. Virtuaalinen konelaboratorio (englanniksi VML, virtual machine laboratory) on konejärjestelmän simulointia, diagnostiikkaa, dokumentaatiota ja toimivan laitteen rakennetta visualisoiva ohjelmisto [22]. VML-ohjelmistoja käytetään, koska ne auttavat hahmottamaan mekatronisen laitteen toimivana kokonaisuutena [22]. Tarve järjestelmän ymmärtämiseksi on selvä: mekatronisen koneen suunnitteluprosessi yhdistää eri suunnittelun osa-alueiden tuottaman suunnitteluinformaation yhdeksi toimivaksi laitteeksi, jonka ymmärtäminen vain yhden suunnittelun osa-alueen näkökulmasta on mahdotonta.

VML:n tarjoamasta toiminnallisuudesta olisi suurta apua jo mekatronisen suunnitteluprosessin aikana. Tällöin suunnittelijoiden tuottamasta dokumentaatiosta rakentuisi toimivan koneen virtuaaliprototyyppi; suunnittelija voisi suunnitteluohjelmilla tuottaa sekä muokata suunnitteluinformaatiota ja katsella rakentuvaa järjes-

telmää virtuaalinäkymien avulla. Tämä helpottaisi eri teknologioiden yhdistämistä suunnittelun aikana, joka on usein mekatronisessa suunnittelussa ongelmallista. Samalla eri alojen suunnittelijat saisivat yhteisnäköyksen suunnitteluun ja kommunikointityövälineen. VML on siis tietokoneavusteinen yhteisöllinen työväline, jolla parannetaan suunnittelua auttamalla suunnittelijoita sovittamaan tekemäänsä työtä yhteen.

Työssä on tavoitteena ensinnäkin tutkia, 1) millainen on hyvä suunnittelua tukeva VML-järjestelmä ja toiseksi tutkia, 2) miten tällainen VML-järjestelmä kannattaa toteuttaa. Lopputuloksena on siis käsitys VML:stä suunnittelun tukityövälineenä, tietoa sen ominaisuuksista ja arvio käytetyistä toteutusteknologioista. Näihin tuloksiin päästään kirjallisuusselvityksellä, asiantuntijahaastattelulla, suunniteltujen ominaisuuksien prototypoinnilla ja teknologioiden toimivuuden arvioinnilla. Kirjallisuusselvityksen perusteella laaditaan kysely, johon koneensuunnitteluprosessin asiantuntijat vastaavat työpajojen yhteydessä. Haastattelutulokset kvantifoidaan sekä analysoidaan ja niiden avulla saadaan käsitys siitä, millainen VML on suunnittelun tukityövälineenä. Keskeisiä ominaisuuksia toteutetaan viimeisimpien ohjelmistoteknologioiden ja käytänteiden avulla tutkimusprototyypin. Tämän tutkimusprototyypin toteutettuja ominaisuuksia arvioidaan Multiple-criteria decision-making -vertailumenetelmällä.

Diplomityö on osa Semogen-hanketta, joka on Tampereen teknillisen yliopiston Smart Simulators -tutkimusryhmän tutkimushanke, jota toteutetaan yhdessä teollisuuden kumppanien kanssa. Diplomityö on tehty matematiikan laitoksen Intelligent Information Systems Laboratory:ssä. Semogen-hankkeessa tutkitaan virtuaalisten konelaboratorioiden tuotantomenetelmiä semanttisen mallinnuksen avulla ja niiden käyttöä suunnittelun tukemisessa [42]. Tavoitteena on tuottaa VML mahdollisimman automatisoidusti sekä tarjota mahdollisuus tarkastella toimivan koneen sisäistä rakennetta ja toimintaa jo koneen suunnittelun aikana [44]. Semogen-hanke jakautuu kaksiosaiseksi hankkeeksi, jonka ensimmäinen osa (Semogen 1) päättyi 2011 vuoden lopussa. Siinä selvitettiin konelaboratorion tuotantomenetelmiä, kuten datan prosessointia ja semanttista mallinnusta konelaboratoriota varten. Hankkeen toinen osa (Semogen 2) päättyi alkuvuodesta 2013. Toisen osan päätutkimusaiheita on muun muassa VML konejärjestelmän suunnittelun tukena. Diplomityössä esiteltävä VML:n datalähtöinen tuottaminen yhteissuunnittelun tarpeisiin on osa Semogen 2-hanketta.

Tutkimuskysymysten asetteluun, visiointiin ja prototyypin toteuttamiseen on osallistunut koko tutkimusryhmä. VML:n tutkimusprototyyppi perustuu Semogen 1-hankkeessa saatuihin tuloksiin ja osittain samaan toteutukseen kuin hankkeen ensimmäisen osan Semoplayer-prototyyppi [44]. Merkittävin diplomityössä rakennettuun prototyypin suoraan Semoplayerista otettu osa on simulaatio ja sen tietoliiken-

neyhteyskomponentti. Sen sijaan erityisesti datalähtöiset VML-näkymäsovellukset semanttiselle tietomallille on ohjelmoinut diplomityön tekijä. Lisäksi osa toteutetuista adaptereista, osa putkilinjasta sekä toimintojen hallintaan tähtäävä toteutus on diplomityön tekijän suunnittelemaa ja toteuttamaa.

Osa diplomityön työstä - erityisesti keskeisimmät tulokset - julkaistaan osana Semogen-hankkeen loppuraporttia. Näitä tuloksia hyödynnetään myös tutkimusryhmän julkaisuissa.

Hankkeen ensimmäisen vaiheen tulokset osoittavat, että lisäämällä suunnittelun aikana digitalisoidun tiedon määrää, suunnittelutiedon yhteensopivuutta ja erityisesti lisäämällä sen semanttista koneluettavuutta saadaan tuotettua konejärjestelmän suunnittelua tukevia näkymiä suoraan suunnitteluaineistosta erilaisten sovittimien ja generointiputkilinjojen avulla [44]. Samalla havaittiin itse suunnitteluprosessissa parannustarpeita kohti semanttisempaa tietojenkäsittelyä [36]. Nämä havainnot ovat lähtökohta Semogen-hankkeen toiselle puoliskolle [42]. [34]

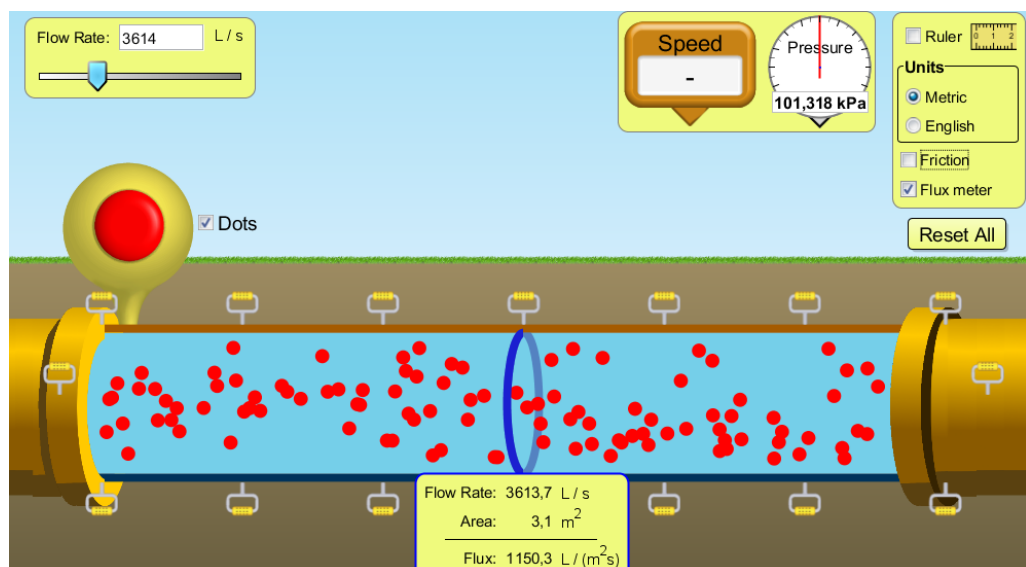
Diplomityössä tutkitaan kartoituksilla, työpajoilla ja kyselyillä VML:n hyötyjä sekä käyttötapauksia suunnittelun tukityövälineenä. Lisäksi tarkastellaan poralaitteen puomin suunnittelun tukemista yhteistoiminnallisilla VML-näkymillä prototyypiohjelmiston avulla.

Teorialuvussa esitellään keskeiset käsitteet ja niiden väliset yhteydet. Luvussa tarkastellaan, mitä työtä Semogen-hankeessa on jo tehty suunnittelun tukemiseksi VML-näkymillä, millaista on tietokoneavusteinen yhteisöllinen työ, millaisia VML-järjestelmiä on toteutettu sekä niiden toteutustapoja, miten mekatronisia laitteita suunnitellaan ja millaisilla tekniikoilla voidaan tuottaa datalähtöisesti sekä automatisoidusti VML suunnitteluinformaatiosta. Teorian jälkeen esitellään tarkemmin tutkimuskysymykset ja -menetelmät. Tuloksissa esitellään saadut havainnot siitä, miten VML:llä voitaisiin käyttää suunnittelun tukemiseksi. Lisäksi esitellään ja arvioidaan prototyypin toteutustavan soveltuvuutta sekä esitetään yleisempi käsitys siitä, miten VML kannattaa toteuttaa suunnittelun tukityövälineeksi. Lopuksi esitetään jatkokehitysideoita ja yhteenveto tämän työn sisällöstä.

2. VIRTUAALIKONELABORATORIOT

Virtuaalilaboratorio (englanniksi VL, virtual laboratory) on tietokoneella luotu laboratorioympäristö sellaisten simuloitujen tapahtumien esittämiseksi, joita voitaisiin tarkastella oikeassa testiympäristössä. Yleisiä vaatimuksia hyvälle virtuaalilaboratorioille ovat uusien teknologioiden ja standardien hyödyntäminen, reaaliaikasisimulaatio, näkymien jakelu tietoverkossa, alustariippumattomuus, edullinen toteuttaa, jatkokehityskelpoisuus sekä sisällön muokkaamisen tuki, asynkronisuus, tarkoituksenmukaisuus, tuki kieliversioinnille, kehittäminen käyttäjäseurannan avulla, modulaarinen toteutus, interaktiivisuus, tietosisällön kommentointi, hakutoiminnallisuus, simuloinnin kontrollointi ja käyttäjän laatimien testien esittäminen. [16] Vaikka nämä ominaisuudet ovat koostettuja opetuskäyttöön tarkoitetuista VL-järjestelmistä, niin ne ovat hyvin yleispäteviä kaikille VL-järjestelmille.

Esimerkkinä VL-järjestelmä fyysikaalisten ilmiöiden opetukseen on PhET (Physics Education Technology), joka tarjoaa mahdollisuuden tarkastella fyysikaalisia ilmiöitä simuloitussa opetusympäristössä. PhET:n avulla voidaan tarkastella ja havainnollistaa mm. voiman, paineen, nosteen, liikkeen ja konsentraation käyttäytymistä luonnossa animoidussa ja interaktiivisessa ympäristössä (kuvassa 2.1) [37].

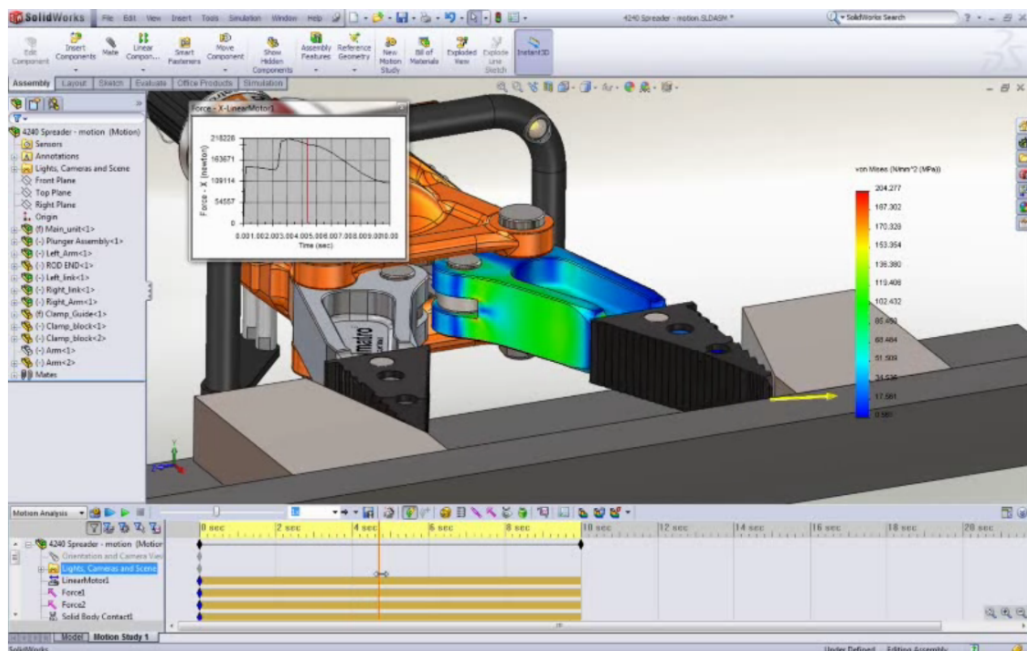


Kuva 2.1: PhET virtuaalilaboratorion avulla voi tarkastella simuloituja luonnonilmiöitä, kuten kuvassa virtauksen tarkastelu.

Eräs VL-järjestelmä on virtuaalinen konelaboratorio (englanniksi VML, virtual machine laboratory), joka on konejärjestelmän simulointia, diagnostiikkaa, dokumentaatiota ja toimivan laitteen rakennetta visualisoiva ohjelmisto. Matemaattisiin malleihin perustava reaaliaikaisimulaatio ohjaa laitteen vuorovaikutusta ja realistista käyttäytymistä.

Esimerkiksi Dassault Group:n rakentama Dassault Systems -järjestelmä toteuttaa VML:n ominaisuuksia. Dassault Systemes on tuotteen elinkaaren hallintajärjestelmä, jonka avulla voidaan tarkastella konelaitteen koko elinkaarta virtuaaliympäristössä. Dassault systems tarjoaa 3D-näkymiä esimerkiksi suunnittelun, tuotannon, markkinoinnin, huollon ja lopulta laitteen kierrätyksen tarpeisiin. [47]

Kuvassa 2.2 on Dassault Systemes:n 3D-suunnitteluohjelmisto Solidworks 3D CAD, jolla voidaan simuloida 3D-mallien avulla toimivan laitteen käyttäytymistä. Virtuaaliprototyypointi on toteutettu hyvin pitkälle Solidworksin 3D-malleihin. [48]

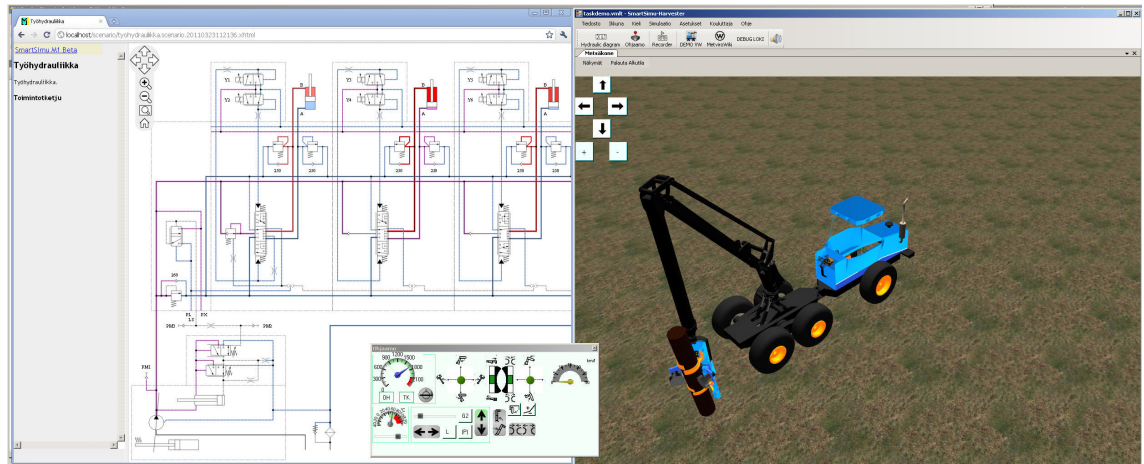


Kuva 2.2: Solidworks 3D CAD:n reaaliaikainen virtuaaliprototyypitestausta ja mittaukset.

VML auttaa hahmottamaan laitteen kokonaisuutena. Tarve on selvä, koska mekatronisissa järjestelmissä mekaniikka, hydraulikka, elektroniikka ja tiedonsiirtoväylät muodostavat yhtenäisen toimivan laitteen. Tätä kokonaisuutta ei voi ymmärtää vain yhden suunnitteluosa-alueen näkökulmasta. Siksi VML:ssä yhdistetään suunnittelutieto kokonaisuudeksi, jotta olisi mahdollista ymmärtää paremmin eri koneen suunnittelun osa-alueiden yhteistoimintaa.

Esimerkkinä tästä on mekatronisen laitteen koulutuskäyttöön kehitetty M1-teknologia ja sitä käyttävä Metviro-prototyyppi [22]. M1-teknologialla tarkoitetaan yhdistelmää teknisiä ratkaisuja, joilla voidaan tuottaa toimiva VML-järjestelmä. Metviro-prototyypin jälkeen M1-teknologiaa on tarkennettu ja laajennettu Semogen-hankkeessa, jonka ensimmäisen vaiheessa tehty Semoplayer-prototyyppi esittelee parempia käytänteitä luoda virtuaalinen konelaboratorio [44].

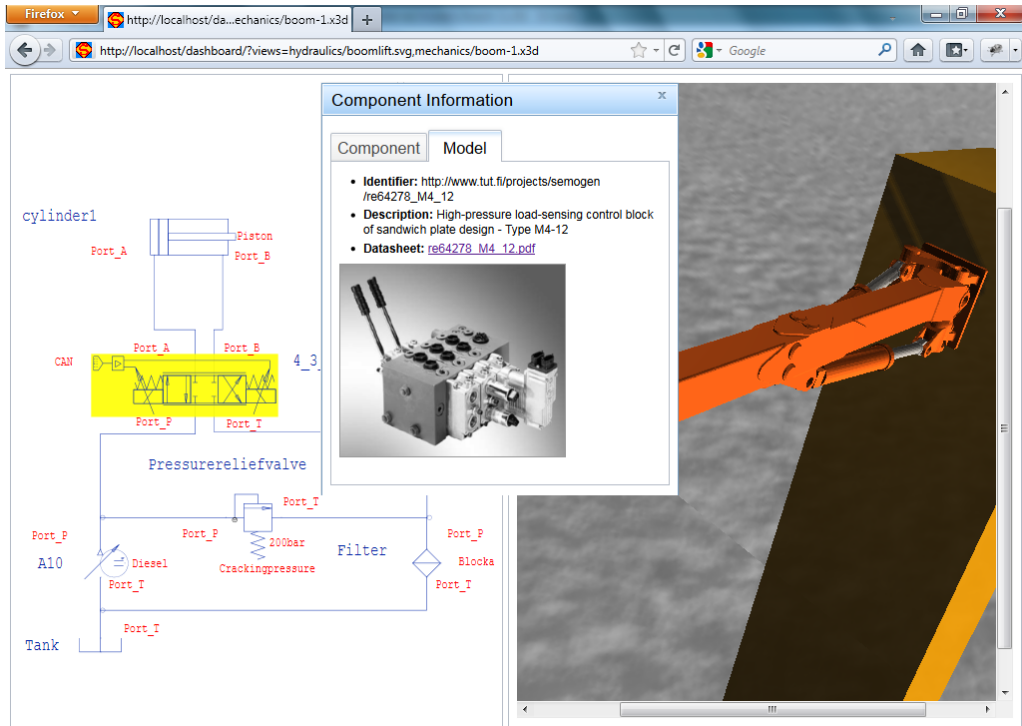
Metviro-prototyyppi (kuva 2.3) virtuaalinen oppimisympäristö metsäkoneharvesterille. Kaaviot ja laitteen 3D-mallit liikkuvat simulaation avulla ja käyttäjä voi ohjata laitetta joystickeilla.



Kuva 2.3: Yleisnäkymä Metviro-ohjelmiston metsäkoneen 3D-mallista ja vasemmalla sen työhydrauliikkajärjestelmä. Kaavio ja malli liikkuvat reaaliaikaisimulaation avulla.

Metviro-prototyyppi on tarkoitettu opetuskäyttöön. Sen avulla on mahdollista ymmärtää helposti suunnittelukaavioiden ja toimivan koneen mekaniikan välinen yhteys. Tästä esimerkkinä kuvassa 2.3 esitetään tapa esittää 3D-mallina olevan laitteen puomin noston ja hydrauliikkaavion yhteys toisiinsa. Kaavioissa ja mallissa näkyvä liike perustuu dynaamiseen simulointiin ja se tapahtuu reaaliaikaisesti käyttäjän ohjauksen perusteella tai toistamalla nauhoitettu toimintoketju.

Semoplayer-prototyyppi (kuvassa 2.4) taas on tuotettu suunnittelun tukityövälineeksi. Semogen-hankkeen ensimmäisen osan Semoplayer-prototyyppi tähtää semanttisen mallinnuksen avulla parantamaan VML:n tuotantomenetelmiä ja ohjelmistoarkkitehtuuria.¹



Kuva 2.4: Semoplayer-ohjelmiston virtuaalikonelaboratorionäkymä. Näkymä näyttää suunnittelijalle suunniteltavan simulaatio-ohjatun puomin ja hydraulikaavion, josta voidaan klikkaamalla tutkia komponentteja. Käyttöliittymä on kokonaan selaimessa.

Teknisesti Semoplayer ratkaisi yhden yhtenäisen selainkäyttöliittymän. Käyttäjä voi avata VML-ohjelmiston selaimella, joka saa tarvittavat resurssit ja simulaatiotiedon verkossa olevalta palvelimelta. Lisäksi näkymien luonnissa hyödynnettiin suunnitteluaineistosta generoidun semanttisen mallin antamia linkityksiä.

¹Lisätietoa Semogen-hankkeesta, julkinen wikisivu <http://wiki.tut.fi/SmartSimulators/Semogen> (27.06.2012)

2.1 VML:n käyttömahdollisuudet

Tyypillinen virtuaalinen konelaboratorio tarjoaa näkymän laitteen toimintoihin ja rakenteeseen. Nämä VML-ohjelmistot ovat toteutettuja tiettyihin käyttötarpeisiin ja niiden toiminnallisuus, visualisoinnit ja simulointi ovat toteutettu riippuen käyttötarkoituksesta. Käyttötarkoituksen perusteella VML-ohjelmistoja voi jaotella kolmeen eri käyttökategoriaan, jotka ovat

1. suunnittelutyön tukeminen,
2. opetuskäyttö ja
3. koneen käyttöopas ja älykäs dokumentaatio [27].

Suunnittelun tukityövälineenä VML auttaa eri alojen suunnittelijoita hahmottamaan suunniteltavaa konetta ja tarjoaa mahdollisuuden tarkastella reaaliaikaisimulaation avulla toiminnallista prototyyppiä. Suunnittelijat voivat virtuaaliprototyypin avulla testata, optimoida, validoida ja ulkoistaa suunnittelutyötään.

Virtuaalisena oppimisympäristönä VML tarjoaa mahdollisuuden oppia ja ymmärtää koneen toimintaa tarkastelemalla mekatronisen koneen toimintoketjuja. Virtuaalinen oppimisympäristö rohkaisee oppijaa kokeilemaan ja harjoittelemaan aidontuntuisessa ympäristössä, jossa virhekäytöstä ei seuraa vastaavia ongelmia kuin oikeassa maailmassa voisi seurata [27]. Tällaisesta VML:stä esimerkkinä on M1-tekniikoilla rakennettu Metviro-oppimisympäristö metsäkoneharvesterille. Metvirossa opiskelija pystyy itsenäisesti tutustumaan 3D-mallinnettuun harvesteriin, jossa on realistinen ohjaus joystick-ohjaimien avulla. Toisaalta opiskelija voi myös ajaa jonkun valmiin toimintoketjun ja tarkastella sen toteutumista laitteessa. Koneen toimintaa voi tarkastella 3D-kappaleiden liikkeenä myös laitteen sisältä ja toisaalta dynaamisista hydraulikka-, CAN- ja sähkökaavioista. Ohjelmistossa on myös mittalaitteet, joilla voi tarkkailla esimerkiksi hydraulikan painetta ja sähkön jännitettä laitteessa.

Mekatronisen koneen käyttöoppaana virtuaalinen konelaboratorio on interaktiivinen kunnossapito-ohjekirja ja komponenttien etsintään sekä tarkasteluun soveltuva näkymä. Tällaista VML-ohjelmistoa voi käyttää esimerkiksi huoltoa suorittava käyttäjä, jolloin ohjelmisto avustaa huoltotoimenpiteiden suorittamista oikein.

2.2 VML:n toiminnot

Eri käyttötapauksiin suunnitellut VML-järjestelmät jakavat tiettyjä yhteisiä ominaisuuksia (taulukossa 2.1), jotka on havaittu M1-teknologiaa hyödyntävistä järjestelmistä [44].

Taulukko 2.1: Yhteiset ominaisuudet eri käyttötarkoituksiin tarkoitetuille VML-järjestelmille M1-teknologioita mukaillen [44].

#	Ominaisuus ja sen kuvaus.
1.	Konejärjestelmän muutosten tuki. Mahdollisuus muuttaa ohjelmiston käyttämää konejärjestelmää vaivattomasti.
2.	Dynaaminen reaaliaikasmulaatio. Järjestelmän toiminnan simulointi ja simuloinnin liittäminen koneen malliin. Mahdollisesti laitteen ohjauksen tarjoaminen käyttäjälle.
3.	Selainpohjainen käyttöliittymä. VML suoritetaan paikallisessa tai verkossa toimivassa palvelimessa ja siihen kytkeydytään WWW-selaimella, jossa on tarjolla ohjelman käyttöliittymä.
4.	Semanttinen haku. Dokumenttien, komponenttien, toimintojen ja muiden resurssien haku semanttisten hakukriteerien avulla (esimerkiksi etsi jokainen komponentti, joka osallistuu tiettyyn toimintoketjuun laitteessa).
5.	Informaation ja aineiston hyödyntäminen muokattavien skeemamallien avulla. VML yhdistää useista eri osa-alueista koostuvan mekatronisen laitteen kokonaisuudeksi. Yhdistämisessä tarvittava skeema on monimutkainen, mutta sitä on pystyttävä hallitsemaan ja muokkaamaan helposti.
6.	Visualisoinnit laitteesta. Kaksi- ja kolmiulotteiset interaktiiviset visualisoinnit konejärjestelmästä yhdistettynä simulaation dynamiikkaan.
7.	Yhteistoiminnalliset näkymät. Koneesta halutaan näyttää eri tekniikoita yhdistäviä näkymiä, kuten esimerkiksi toimintoketjujen seurantaa. Abstraktit käsitteet helpottavat monimutkaisen moniteknisen laitteen ymmärtämistä.
8.	Mittaukset. Mahdollisuus mitata laitteesta muuttumattomia (massa, tilavuus, pituus) ja muuttuvia suureita (paine, paikka, voima) on tärkeä ominaisuus toiminnan seurannassa. Esimerkiksi sylinterin paineen muutoksen seuranta kuvaajana simulaation tuottamien lukuarvojen perusteella.
9.	Aidontuntuisen ohjauskontrollin antaminen laitteeseen. Laitteen aidon ohjaustuntuman tuottaminen graafisen käyttöliittymän, tietokoneen osoitinlaitteiden tai VML-järjestelmään liitetyn oikean vastaavan laitteen ohjauslaitteiston avulla.
10.	Simulaation hallinta. Pelkän realistisen reaaliaikaisen simulaation lisäksi käyttäjä haluaa mahdollisesti muuttaa simulaation toimintaa esimerkiksi esittämällä hidastettuna tietty laitteen toiminto.

M1-teknologioilla on ratkaistu Metvirossa taulukon 2.1 kohdat 2 dynaaminen reaaliaikasisimulaatio, osittain 3 selainpohjainen käyttöliittymä, 6 visualisoinnit laitteesta, 7 yhteistoiminnalliset näkymät, 8 mittaukset, 9 aidontuntuinen laitteen ohjauskontrolli ja 10 simulaation hallinta. Sen sijaan Semogen-hankkeen ensimmäisessä osassa Semoplayer-prototyyppi keskittyi ratkaisemaan myös kohtia 1 konejärjestelmän muutoksien tukea, 3 täysin selainpohjaista VML-käyttöliittymää, 4 semanttista hakua ja 5 informaation hallintaa muokattavien skeemojen avulla [44].

Diplomityössä selvitetään VML:n käyttöä suunnittelun tukityövälineenä ja karotetaan ominaisuuksia, joita VML:ssä tarvitaan suunnittelun tukemiseksi. VML-tutkimusprototyyppiin toteutetaan yhteissuunnittelua varten erilaisia ominaisuuksia. Toteutettua prototyyppiä arvioidaan yleisten VL-vaatimusten toteutumisen kannalta. Eräs tällainen vaatimuslista sisältää seuraavat vaatimukset: uusien tekniikoiden ja standardien hyödyntäminen; reaaliaikasisimulaatio; näkymien jakelu tietoverkossa; alustariippumattomuus; edullinen toteutus; jatkokehityskelpoisuus ja sisällön muokkaamisen tuki; asynkronisuus; tarkoituksenmukaisuus; tuki kieliversioinnille; kehittäminen käyttäjäseurannan avulla; modulaarinen toteutus; interaktiivisuus; tietosisällön kommentointi; hakutoiminnallisuus; simuloinnin kontrollointi ja käyttäjän laatimien testien esittäminen [16]. Nämä ominaisuudet ovat hyvin yleispäteviä sekä lähellä VML:n M1-teknologian mukaisia ominaisuuksia.

3. NÄKÖKULMIA SUUNNITTELUUN

3.1 Tietokoneavusteinen yhteisöllinen työ

Tietokoneavusteinen yhteisöllinen työ (englanniksi CSCW, computer supported cooperative work) on yleistermi tietotekniikan avulla suoritettavalle ihmisten väliselle yhteistyölle [56]. Eräitä CSCW-järjestelmiä ovat esimerkiksi sähköposti ja wikijärjestelmät, joiden avulla ihmiset voivat työskennellä koordinoitusti maantieteellisesti erillään. Teknisesti, kuten CSCW-järjestelmät yleensä, nämä esimerkkijärjestelmät vaativat tietokoneen, tietoverkon ja tarvittavat ohjelmistot. Järjestelmien toteuttaminen ja käyttöönotto onkin helpottunut internetin ja tietokoneiden yleistyttyä käyttöön kaikkialla.

Motivaatio tietokoneavusteisten yhteisöllisten työvälineiden tutkimiselle ja käytölle on peräisin oletuksesta, että ne tehostavat ja helpottavat ryhmätyöskentelyä, sekä auttavat ryhmää saavuttamaan parempia lopputuloksia. CSCW-teknologiat pyrkivät auttamaan ryhmätyötä neljällä eri tavalla [56, s. 1]:

1. Kommunikointivälineenä mahdollistaen eri paikoissa olevien ihmisten tiedon jakamisen keskenään.
2. Yhteisen työtilan avulla siten, että kaikki voivat tarkastella toistensa tekemää työtä.
3. Yhteisten jaettujen resurssien tarjoajana esimerkiksi aineistotietokannan avulla.
4. Prosessin ohjaajana siten, että ihmiset työskentelevät ryhmässä ja jakavat työtehtäviä.

Tähän jaotteluun on päädytty, kun on haluttu vastata CSCW-järjestelmien avulla organisaatioiden tarpeeseen

- parantaa kommunikaatiota ihmisten välillä;
- lyhentää päätöksentekoaikoja;
- parantaa strategisten päätösten laatua;
- muuttaa nopeasti organisaation rakennetta;

- nopeuttaa siirtymiä uusille markkina-alueille;
- muokata ja luoda tuotteita nopeammin;
- parantaa ihmisten yhteistyötä
- ja tarjota asiakkaalle laadukkaampaa palvelua [56, s. 9].

Ilmeisemmin CSCW vastaa näihin vaatimuksiin, mutta sopivaa toteutettua järjestelmää ei välttämättä ole saatavilla, vaikka organisaatio olisi tunnistanut sen tarpeen. Toisaalta yleiskäyttöiset CSCW-järjestelmät, kuten wikit, sähköposti, pikaviestimet ja versionhallintajärjestelmät, ovat jo arkipäiväisiä työkaluja pienille sekä suurille organisaatiolle ja ryhmille.

CSCW-järjestelmiä voidaan luokitella niiden käyttötavan ja käyttötilanteen mukaan. Kuvassa 3.1 on esitetty malli jäsenellä CSCW-järjestelmiä työskentelyn samanaikaisuuden sekä sen mukaan, että tapahtuuko työskentely paikallisesti vai erillään [7, s. 742].

	Asynkroninen kommunikaatio	Synkroninen kommunikaatio
Eri paikoista	<p><u>Kommunikointi ja koordinointi</u></p> <p>tikettien hallinta wikit email</p>	<p><u>Etäyhteistyö</u></p> <p>pikaviestimet yhteiseditorit videoneuvottelut</p>
Samassa paikassa	<p><u>Jatkuvat tehtävät</u></p> <p>suuret julkiset näytöt vuorotyöohjelmat projektinhallinta</p>	<p><u>Kasvotusten</u></p> <p>videoprojektori yhteinen näppäimistö älytaulut</p>

Kuva 3.1: CSCW:n konseptien esittely. Matriisissa tarkastellaan sitä, tehdäänkö työtä fyysisesti samassa tilassa vai etänä, ja tapahtuuko työskentely samanaikaisesti vai eri aikoihin.

Kuvan 3.1 mallin perusteella sähköposti ja wiki ovat asynkronisia, eli käyttäjä ei koko ajan seuraa muiden toimintaa. Sitä vastoin pikaviestintä taas on synkronista kommunikaatiota, koska osapuolet seuraavat toistensa viestejä reagoiden niihin välittömästi.

Myös VML-järjestelmät ovat tietokoneavusteisen yhteisöllisen työn välineitä, ja niiden ominaisuuksia tarkastellaan diplomityössä kuvan 3.1 nelikenttän avulla.

Tietokoneavusteiset yhteisölliset työvälineet ovat yleisesti omaksuttuja käyttöön esimerkiksi yhteisiä toimisto-ohjelmia vaativassa projektityöskentelyssä. Ihmisten

kommunikointia ja työn koordinoitua avustava yhteistyöväline on esimerkiksi Googlen tarjoama Google Docs -toimisto-ohjelmistopaketti. Google tarjoaa näitä toimisto-ohjelmistoja ilmaisena Software as a Service -palveluna, jota voi käyttää modernilla selaimella. Palvelun toimisto-ohjelmat ovat yhteisöllisiä, asynkronisia ja hajautettuja työvälineitä, koska dokumentin näkyvyyttä ja käyttöoikeuksia voi hallita siten, että samaa dokumenttia voidaan tuottaa samanaikaisesti yhteistyönä eri puolilta maailmaa. [18]

3.2 Mekatronisen koneen suunnittelu

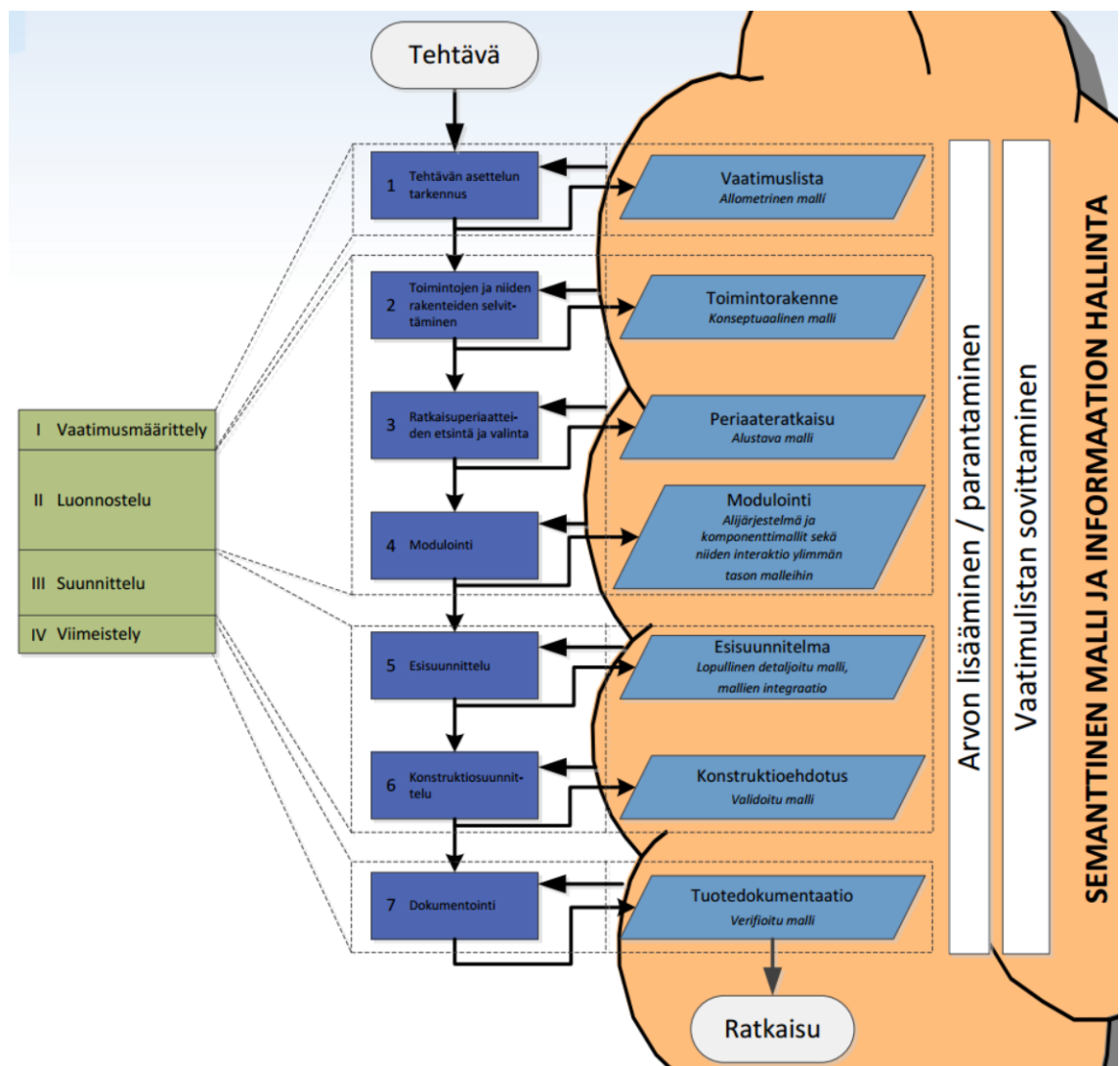
Alkuperäinen määritelmä mekatroniikalle oli mekanismi (meka) yhdistettynä elektroniikkaan (troniikka). Toisin sanoen teknologiamekanismien suunnittelu, jossa elektroniikka yhdistyy saumattomasti mekanismin toimintaan. Mekatroniikan määritelmä on kuitenkin muuttunut ja elää koneensuunnittelun teknologioiden kehityksessä. [10]

Nykyään kuitenkin tietotekniikka, automaatiotekniikka ja systemitekniikka ymmärretään osaksi mekatroniikkaa [10]. Toisaalta esimerkiksi TTY:lläkin käytetty perusoppikirja määrittelee ”Mekatroniikka: oppi tietotekniikan hyväksikäytöstä koneissa ja laitteissa.” [2, s. 1], jota voidaan pitää liian suppeana määritelmänä, koska mekatroniikalla ja perinteisellä koneenrakennuksella ei ole nykyisin mitään selvää rajaa. Näistä syistä tässä diplomityössä ei tehdä eroa mekatroniikalle ja perinteiselle koneenrakentamiselle tai -suunnittelulle.

3.2.1 Suunnitteluprosessi

Mekatronisen tuotteen kehittäminen asettaa vaatimuksia tehtävänasetteluun, henkilöiden väliseen kommunikaatioon, tietokoneavusteisiin työkaluihin ja projektinhallintaan. Yleisesti tuotekehityksessä käytetään saksalaista VDI 2221:n metodiikkaa, joka jakautuu tehtävänasetteluun, luonnosteluun, kehittämiseen ja viimeistelyyn. Malli ei kuitenkaan sellaisenaan toteudu vaiheesta toiseen teollisuudessa, koska jo tehtyjä ratkaisuja joudutaan muuttamaan uusien havaintojen tullessa esiin suunnittelun aikana; toteutuva suunnitteluprosessi on iteratiivinen. [2, s. 13]

Kuvassa 3.2 Semogen-hankkeessa tuotettu VDI-pohjainen semanttinen suunnitteluprosessimalli, jota esiteltiin Semogen-hankkeen aikana työpajoissa.



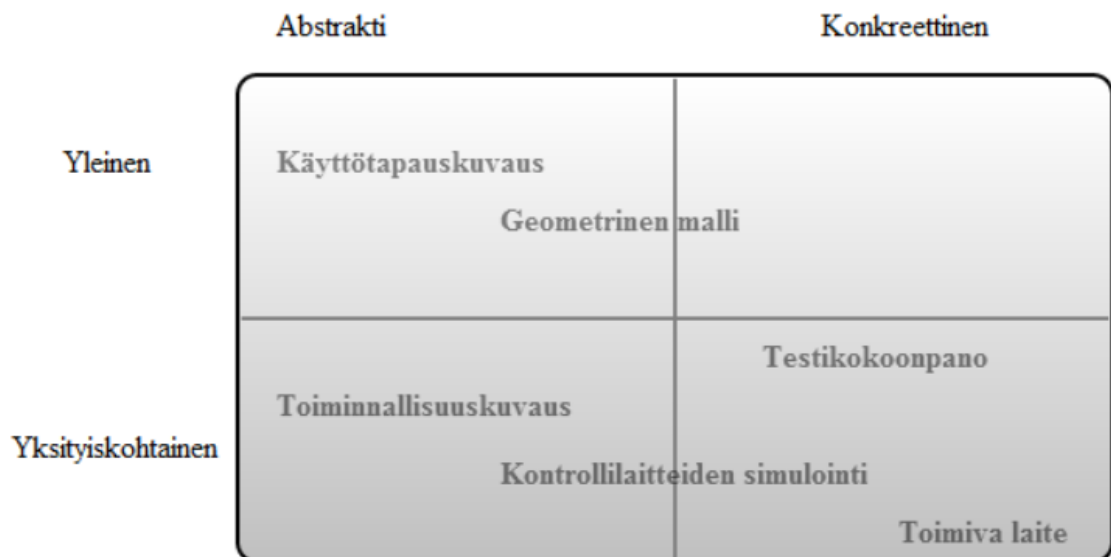
Kuva 3.2: Semogen-hankkeessa tuotettu VDI-metodiikan mukaisesti etenevä semanttinen suunnitteluprosessi. Prosessikuva on Semogen-hankkeen työpajamateriaalista.

Kuten mikään mukaan suunnittelumalli, paljon käytetty VDI-malli ei ratkaise kaikkia suunnittelun ongelmia. Eräs ongelma on esimerkiksi se, miten löydetään

oikeat asiakasvaatimukset ja varmistetaan, että ollaan toteuttamassa lopputuotteen juuri näitä oikeita vaatimuksia. Suunnittelumalleja onkin kritisoitu siitä, että ne eivät aina onnistu tuottamaan haluttuja asiakasvaatimuksia lopputuotteen ja olettavat liian varhaisessa vaiheessa suunnittelua lukittavia varmoja vaatimuksia [28].

Tiedon ja mallien näkökulmasta suunnitteluprosessi etenee niin, että abstraktista ja luonnostason tiedosta edetään kohti konkreettista ja yksityiskohtaista suunnittelutietoa. Alussa on abstrakteja koneen ominaisuuksia - käyttötapaukset ja toiminnot - sekä luonnostason suunnitelmia, joita lähdetään tarkentamaan. Lopullinen konkreettisin ja yksityiskohtaisin lopputulos on rakennettu kone ja siihen liittyvä dokumentaatio [28].

Kuvassa 3.3 on erilaisia malleja koneesta sijoitettuna niiden konkreettisuuden ja tarkkuuden mukaan. Suunnitteluprosessissa suunnittelutieto jalostuu kuvan vasemmasta ylänurkasta kohti oikeaa alanurkkaa.



Kuva 3.3: Suunnitteluprosessissa syntyy erilaisia malleja suunniteltavasta koneesta. Suunnittelu etenee abstraktista konkreettiseen ja tarkempaan malliin koneesta.

Kuvan 3.3 mukaisesti voidaan VDI:n vaiheet tehtävästä toimivaan koneeseen ajatella kulkevan vasemmasta ylänurkasta oikeaan alanurkkaa. Suunnittelutyössä lopputuloksena on dokumentaatio ja mallit sellaiselle tarkkuudella, että niiden avulla voidaan rakentaa toimiva laite.

3.2.2 Suunnittelutyövälineet

Mekatronisessa suunnittelussa tietokoneavusteinen suunnittelu eli CAD (englanniksi computer-aided design) tapahtuu pääosin teknologia-aloittain eri suunnitte-

luohjelmilla. Esimerkiksi seuraavien teknologia-alojen CAD-ohjelmat toimivat erillään toisistaan: mekaniikkasuunnittelu (MCAD), sähkösuunnittelu (ECAD), CAN-väyläsuunnittelu (CCAD) ja hydraulikkasuunnittelu (HCAD). Ikävä kyllä CAD-ohjelmistot eivät tue suunnittelutiedon siirtämistä suunnittelijalta toiselle tietokoneavusteisesti. Näin ollen suunnittelutieto siiloutuu eri suunnittelun osa-alueiden sisälle jo ohjelmistojen tasolla. [58]

Tuotetiedon hallintaohjelmistot eli PDM-ohjelmistot (englanniksi product data management) yrittävät ratkaista mekatronisen suunnittelutiedon hallinnan ongelmia. Eri suunnitteluohjelmat tallentavat tuotetun suunnittelutiedon dokumentteina PDM-järjestelmään. PDM-järjestelmissä on tiettyä suunnitteluprosessia varten suunnitellut paikat tiedostoille ja mahdollisuus lisätä metatietoja tiedoston lisäämisen lisäksi. Näin tiedolle voidaan tuottaa erilaisia näkymiä ja hakuja esimerkiksi käyttäjien käyttöoikeuksien mukaan. Tiedostoille on käytössä luokitteluita, joiden mukaan tiedostot lisätään ja esitetään. Tiedostoja voi muokata, joten työnkulkua voidaan tarkastella ja suunnitella PDM:ssä. Yhteissuunnittelun kannalta PDM tarjoaa yhteisen paikan suunnittelutiedostojen tallennukseen ja tarkasteluun. [15]

Kuitenkin vaikuttaa siltä, että PDM-järjestelmiä käytetään sellaisella tavalla, että ne karkeasti ottaen hallitsevat tiedostotasolla tuotetiedon hallinnan. Ne kertovat eri tiedostojen luokituksia ja suhteita toisiinsa sekä hyödyntävät tiedostojen mukana lisättyä metatietoa. Varsinainen täsmällinen suunnittelutieto on kuitenkin tiedostojen sisällä. Semogeen-hankkeessa tarkastellaan suunnittelutiedon hallinnan hyötyjä: jotta esimerkiksi VML voi hyödyntää täsmällistä suunnittelutietoa (esimerkiksi komponentti kaaviossa), täytyy suunnittelutieto mallintaa ja hallita luokitellusti ja linkitetysti metatietoineen.

On myös tärkeää tiedostaa, että varsinaisten mekatronisen suunnittelun tietokoneavusteisten työvälineiden lisäksi yrityksissä on usein käytössä muita yleisiä suunnittelua avustavia tietokoneohjelmia, kuten sähköposti, tikettijärjestelmä, wiki ja pikaviestimiä. Näitä välineitä ei kuitenkaan käytetä tuottamaan sellaista suunnittelutietoa, jota voitaisiin liittää osaksi muuta suunnitteludokumentaatiota. Tässä diplomityössä yritetään liittää VML-järjestelmään myös sellaisten suunnittelua tukevien tietokoneohjelmien tietoa, joiden tietoa ei tällä hetkellä liitetä PDM-järjestelmiin.

3.2.3 Suunnittelutiedon tallennusmuodot

Suunnitteluohjelmat tallentavat suunnittelutiedon tiedostoihin usein valmistajasi-
donnaaisissa formaateissa. Lisäksi suunnitteluohjelmat keskittyvät tiedon visuaalisen muodon tallentamiseen esimerkiksi kaavion tulostamista ajatellen, vaikka oleellisempaa olisi tallentaa kaavion kuvaama suunnittelutyö. Nyt suunnittelutyö siiloutuu esimerkiksi PDF-dokumentin sisälle sellaiseen muotoon, että tietokone ei osaa tulkita sitä suunnittelutiedoksi; suunnittelutieto jää tietokoneohjelmilta hyödyntä-

mättä. [42]

On mielekästä tarkastella tietosiilojen avaamista. Yhteissuunnittelun ja VML:n kannalta voidaan etsiä etuja, jotka semanttisen webin avoimet teknologiat tarjoavat suunnittelutiedon yhdistelyyn ja hyödyntämiseen. Visiona on saada suunnittelun lopussa tulostettavaksi tarkoitetun suunnitteludokumentaation lisäksi yhtenäinen linkittynyt kokonaismalli suunnittelutyöstä, jossa tiedostojen sisältämä tieto - kuten kaavion kuvaama komponentti - on linkittynyt koneluettavasti muiden dokumenttien suunnittelutietoon. Yhteissuunnittelu ja VML voivat hyödyntää suunnittelutietoa vasta, kun dokumenttien sisältämä tieto hallitaan, luokitellaan ja linkitetään metatietoineen onnistuneesti keskenään.

Suunnittelutiedon tallennus yhtenäiseen muotoon on hankala ongelma, koska eri ohjelmistojen valmistajilla ei ole liiketoiminnan näkökulmasta tarvetta tarjota yhtenäisiä ohjelmistorajapintoja tai tallennusformaatteja. Lisäksi tiedon mallinnustavasta sopiminen on hyvin haastava ongelma.

Eräs askel suunnitteluohjelmistojen välisen tiedonsiirron toteuttamiseen on avoin OSLC-yhteisö (Open Services for Lifecycle Collaboration), joka pyrkii usean ison teollisuuden toimijan kanssa sopimaan avoimen muodon, kyselytavan ja jakelumahdollisuuden eri suunnitteluohjelmistojen väliselle suunnittelutiedolle. OSLC:ssä osallistujat suunnittelevat ohjelmistotuotannon näkökulmasta alkuaskelta hyödyntää eri ohjelmistojen suunnitteluohjelmien tuottaman tiedon yhdistelyä webin teknologioiden avulla. [39] Vaikka ratkaisuja haetaan ohjelmistotuotannon käyttämiin suunnitteluohjelmistoihin, niin ratkaisumalli on toimiva myös koneensuunnitteluohjelmistojen tiedon yhdistelyn ongelmiin.

Semogen-hankkeessa on myös vastaavasti tutkittu ja testattu yksinkertaista tapaa lähteä ratkaisemaan suunnittelutietojen julkaisun, hallinnan, löytämisen ja yhdistelyn ongelmaa. Lähestymistapana on ollut mallintaa ja julkaista aivan suunnittelun keskeisin komponentti-, laite- tai järjestelmätiedon sisältävä datalehtitieto koneluettavasti. Datalehtien tietoa käytetään suunnittelussa tiedonvaihtoon suunnittelijoiden ja muiden prosessin jäsenten välillä. Tieto on kuitenkin yleensä tallennettu lähinnä tulostusta ja katselua varten PDF-tiedostoihin. Ratkaisuna on ollut tarjota datalehden sisältämä tieto myös siten, että tietokoneohjelmat voivat hyödyntää sitä mahdollisimman hyvin.

Semanttisen webin teknologiat ovat osoittautuneet Semogen-hankkeen ensimmäisessä osassa mielekkäästi sovellettaviksi koneen mekatroniselle suunnittelutiedolle juuri tietojen mallintamisessa ja yhdistelyssä. Tällöin tiedot eivät ole pelkästään digitaalisesti tallennettuja, vaan siten, että niiden tietosisältöön voidaan viitata ja tieto on tietokoneohjelmien hyödynnettävissä.

3.3 Suunnittelumalleja

Luvussa esitellään lyhyesti käsitteet bottom-up, top-down ja iterointi, jotka esiintyvät monissa suunnittelumalleissa. Nämä ovat siten myös keskeisiä käsitteitä semanttisessa suunnitteluprosessissa.

Informaatiota voidaan prosessoida hahmottamalla kokonaisuus yläkäsitelmallista kohti yksityiskohtaisempia alakäsitelmalleja tai sitten lähtemällä alakäsitelmalleista ja etenemällä kohti yläkäsitelmallia. Top-down (ylhäältä alas) ja bottom-up (alhaalta ylös) lähestymistavoilla tarkoitetaan vaihtoehtoisia tapoja hahmottaa konejärjestelmää.

- Top-down: alkaen yläkäsitelmallista ja edeten kohti alijärjestelmien malleja.
- Bottom-up: alkaen alijärjestelmistä ja edeten kohti yläkäsitelmallia.

Iteroinnilla taas tarkoitetaan konejärjestelmän suunnitteluratkaisujen testaamista koko suunnittelun ajan sekä mahdollisuutta palata suunnittelemaan uudelleen jo tehtyä työtä. Tällöin voidaan varmistaa valittujen suunnitteluratkaisujen vievän suunnittelua oikeaan suuntaan.

Koska mekatronisessa suunnittelussa usein lukitaan suunnittelun varhaisessa vaiheessa tiettyjä teknologiavalintoja ja jaetaan suunnittelu näiden teknologiavalintojen mukaan, niin tämän seurauksena suunnitteluprosessissa on rajoitetut mahdollisuudet parannella varhaisia suunnitteluvalintoja. Lisäksi on tunnistettu mahdollisuus parantaa tätä tilannetta huomalla moniteknistä konejärjestelmää varten teorioita, malleja ja työkaluja, jotka hallitsevat mallinnuksen, analysoinnin, yhdistämisen, simuloinnin ja prototypoinnin. Toivottava lopputulos olisi sellainen kehitysstrategia, joka olisi hahmotettavissa suunnittelun konseptien ja yläkäsitteiden avulla. [1, p. 653]

Tutkimuksen kannalta on mielekästä testata, miten VML-järjestelmällä voitaisiin vastata näihin tunnistettuihin tarpeisiin. Toisaalta yhteissuunnittelu vaatii tietoja yhdisteleviä ylätasoa malleja, jotka pitää tuottaa myös koneluettavaan muotoon, koska tällaisten ylätasoa mallien avulla suunnittelutieto yhdistyy myös ohjelmistoja varten.

3.4 Yhteissuunnittelu

Yhteissuunnittelu (englanniksi co-design) on suunnittelun muoto, jossa kaikki projektin jäsenet ovat yhdessä suunnittelemassa suunnitteluratkaisuja koko suunnitteluprosessin ajan [46]. Yhteissuunnittelu voi ilmetä yrityksessä esimerkiksi siten, että koko tuotteen suunnittelun ajan tuotetta suunnittelevat henkilöt ja tulevat käyttäjät (asiakkaat) kommunikoivat suunnitteluratkaisusta keskenään. [17]

Yhteissuunnittelun käsitteitä on pohtinut filosofi Immanuel Kant yrittäessään määritellä ihmisen tiedon muodostamista. Jotta voi muodostaa järkeviä ongelman asetteluja, täytyy olla hallussa ne mallit ja tiedot, joista ongelma koostuu. Lähtökohtana yhteissuunnittelussa on tarjota tieto eri ihmisten saataville, jotta he osaisivat muotoilla ongelmien ja kysymysten asettelun oikein. Ongelmien ja kysymysten asettelua pohtimalla löydetään lopulta yhteinen näkemys suunnittelukysymyksistä. Tietoteorian näkökulmasta näihin teemoihin Kant pyrkii vastaamaan teoksessaan *The Critique of Pure Reason*. [26]

Vastaavasti mekatronisen koneen suunnittelu yrittää vastata koko suunnittelun ajan kysymyksiin, miten rakentaa kone, joka on asetettujen vaatimusten mukainen. Toisaalta suunnittelussa syntyvä tieto tarkentaa ymmärrystä siitä, mitkä ovat oikeat vaatimukset suunniteltavalle koneelle.

Tässä diplomityössä yhteissuunnittelulla ajatellaan olevan seuraavia piirteitä:

- Yhteissuunnittelu pyrkii tuomaan erilaiset näkemykset ja ihmiset saman ”suunnittelupöydän” ääreen.
- Yhteissuunnittelu ei ota kantaa käytössä oleviin suunnittelumalleihin.
- Yhteissuunnittelun toivotaan parantavan oikeiden suunnitteluvaatimusten ja -ratkaisujen löytämistä.
- Yhteissuunnittelua sovelletaan suunnitteluratkaisuihin koko suunnitteluprosessin ajan.
- Jokaisen osallistujan tulee hahmottaa, miten hänen työnsä vaikuttaa suunniteltavaan kokonaisuuteen.

Täsmällisemmin tässä diplomityössä pohditaan koneensuunnittelun kontekstissa yhteissuunnittelua, jossa erilaisin näkökulmin konetta tarkastelevat ihmiset saavat yhteisen näkymän koneeseen. Käytännössä tämä näkymä auttaa erilaisten mallien avulla konetta rakentavien ihmisten yhteistyötä ja asetettujen vaatimusten toteutumisen tarkastelua yhdessä saman virtuaalisen suunnittelupöydän ääressä. Kun tieto kokonaisuuden suunnittelusta on saatavilla, niin suunnittelijoiden on helpompi tarkastella suunnittelutyön kysymyksiä yhdessä.

Suunnittelun voidaan ajatella olevan silloin onnistunutta yhteissuunnittelua, kun erilaisin näkökulmin konetta tarkastelevat henkilöt päätyvät tarjolla olevaan suunnittelutietoon nojaten samoihin kysymyksiin ja ongelman asetteluihin. Tällainen tilanne olisi esimerkiksi sellainen, että suunnittelutietoa katselmoivat mekaniikkasuunnittelija ja hydraulikkasuunnittelija yhdessä asiakkaan kanssa päätyvät yhteisymmärrykseen siitä, mikä on seuraava oikea iterointiaskelel.

VML:ssä halutaan kuvata monimutkaisen konejärjestelmän eri osajärjestelmien toiminnan yhteistulosta. Näin voidaan hahmottaa eri teknologioiden kokonaisuuden suunnittelu ja ymmärtää paremmin, millaista suunnittelutyötä ollaan tekemässä.

Suunnittelun tukityövälineenä VML on juurikin yhteissuunnittelun tukityöväline, joka tarjoaa yhteisen näkymän - suunnittelupöydän - suunnittelun tuottamaan informaatioon. Ideaalituloksessa voidaan tällöin koettaa vastata siihen, mitä tietoa on olemassa suunnitteluratkaisujen toimivuudesta, ja miten ne vastaavat vaatimuksiin. Lisäksi osataan muotoilla kysymykset siitä, mitä on tehtävä seuraavaksi.

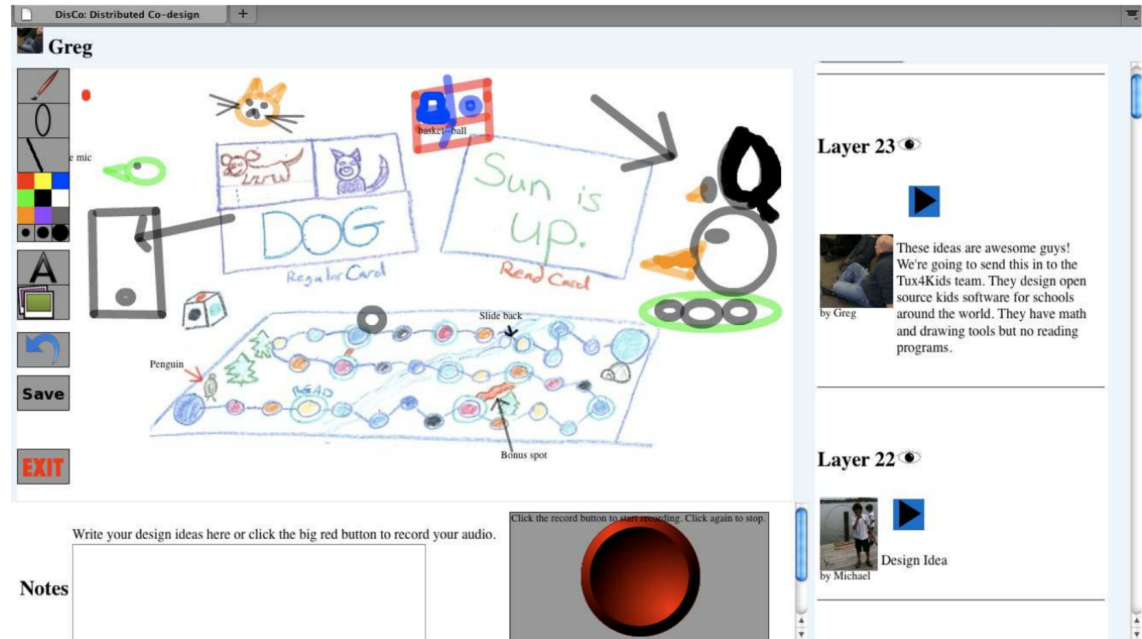
Esimerkiksi yhdyskuntasuunnittelussa on Suomessa pyritty julkaisemaan ihmis-yhdyskuntien rakennetta, käyttöä ja ympäristön tilaa seuraavat aineistot sekä kannustamaan ihmisiä osallistumaan oman elinympäristönsä suunnitteluun hyödyntämällä näitä tietoja. Kuvassa 3.4 on osallistuvan yhdyskuntasuunnittelun ja avoimen datan työpaja, jossa suunnittelua tehdään yhteisen suunnittelupöydän ympärillä.



Kuva 3.4: Esimerkki yhteisen suunnittelupöydän käytöstä yhdyskuntasuunnittelussa. Osallistuva yhdyskuntasuunnittelu - avoin data. Työpaja yhteisen suunnittelupöydän ääressä. Kuvan on ottanut Peter Tattersall <http://www.mahdollista.fi/nettikansan-tapahtumat/2011-tapahtumat/osallistuva/> (07.11.2012).

Yhdyskuntasuunnittelussa on saatu hyviä tuloksia tarjoamalla suunnitteluun osallistuville ihmisille aineistoa visualisoidussa muodossa, kuten esimerkiksi piirroksina, tilastoina, karttoina ja tietokoneella luoduilla kuvilla. Havaintojen perusteella suunnittelutiedon selkeä esittäminen auttaa ongelmien havaitsemisessa ja ratkaisujen löytämisessä yhdyskuntasuunnittelussa. [3]

Ohjelmistoratkaisuilla voidaan toteuttaa hajautettuja asynkronisia yhteissuunnittelun työvälineitä. Esimerkiksi web-teknologioilla toteutettu prototyypiohjelmitto DisCo on asynkroninen yhteissuunnittelun tukityöväline, joka on suunnattu erityisesti lasten saamiseksi mukaan heille suunnattujen tuotteiden suunnitteluun. Hajautettuun yhteissuunnitteluun lapset voivat osallistua selaimella kotoaan. [4] Kuvassa 3.5 on DisCo-prototyypin käyttöesimerkki.



Kuva 3.5: Esimerkkikuva DisCo-yhteissuunnitteluohjelman käyttöliittymästä sen käyttötilanteessa. [4]

DisCo mahdollistaa lasten ja aikuisten välisen yhteissuunnittelun verkossa. Yhdessä asynkronisesti tapahtuva suunnittelu voi olla maantieteellisesti täysin hajautettu [4]. Seuraava esimerkki on lukemispelin suunnittelutilanne, jossa kolme noin 10-vuotiasta lasta suunnittelee peliä eri puolilla Yhdysvaltoja.

”A board game to teach kids to read would be cool...” explains Roberta. Enoch agrees by suggesting, “You should be able to integrate your ebook reader into the game.” Agatha adds “I would do a board game that involves reading and for them to be able to move around”. Oliver suggests a new tack, “I think readers who want a challenge should take a new path” and Alice June adds that players should take a quick reading test so the game knows how well players can read and adjust the game based on it. The design discussion finishes as Max adds, “The board game should be played on a phone and have READ spaces that need you to read from a story when you land on them.” [4]

Vastaavasti VML:n tulisi tarjota yhteissuunnittelulle yhteinen ohjelmistolla toteutettu suunnittelutyöpöytä, jossa suunnittelutieto on saatavilla ja visualisoituna. Näin voidaan avustaa eri näkökulmista konetta tarkastelevia ihmisiä heidän dialektiikassaan ja saavuttaa yhteisymmärryksen kautta parempia suunnitteluratkaisuja.

Mekatronisten järjestelmien tuottaminen vaatii osapuolien yhtäaikaista yhteistyötä. Kuitenkaan suunnitteluohjelmien välillä ei yleensä ohjelmistojen tuella suoraan tapahdu yhteissuunnittelua, vaikka esimerkiksi Dassault Systemes:n tuotteet tähän enenemässä määrin pyrkivätkin; tarve on joka tapauksessa tunnistettu. [51]

4. SEMANTTISEN WEBIN SOVELLUSTEKNIIKAT

4.1 Semanttinen tiedon mallinnus

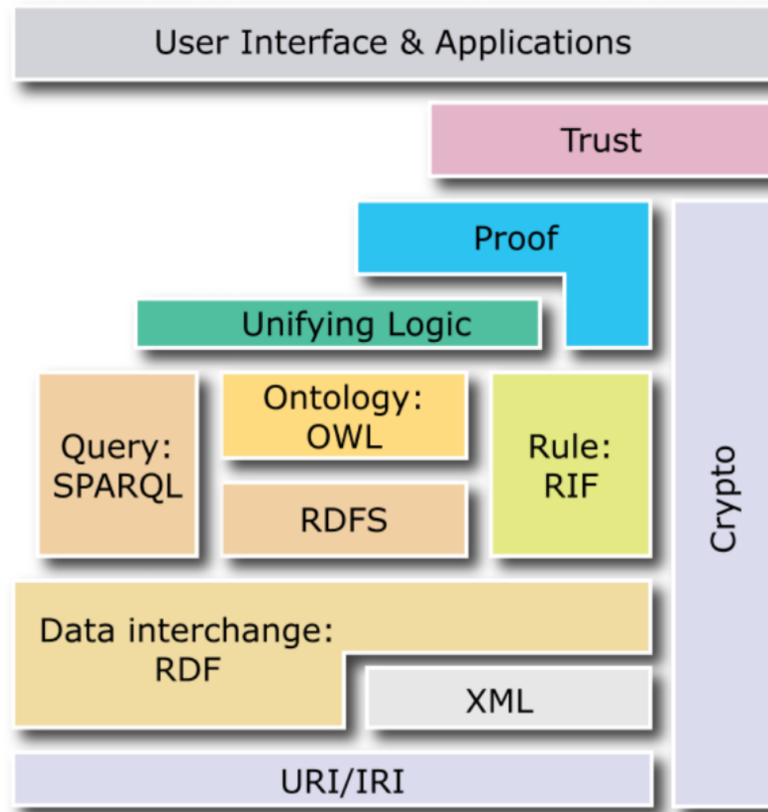
Tiedon semanttinen mallinnus pyrkii mallintamaan tiedon linkitetysti, koneluettavasti ja formaalisti. Keskeinen päämäärä tiedon semanttiselle mallinnukselle on tuottaa loppukäyttäjälle älykkäämpiä ja parempia sovelluksia, jotka ratkaisevat käyttäjän ongelmia entistä paremmin [5].

Älykkäästi toimivasta ohjelmasta esimerkkinä voidaan käyttää esimerkiksi internetin hakukonetta, joka antaa hyvin intuitiivisia vastauksia käyttäjän tekemän haun perusteella suuresta tulkitusta tietomassasta. Semogen-hankkeessa vastaavasti koneen tuotantomenetelmiä kehitetään lisäämällä suunnittelutiedon koneluettavuutta semanttisella mallinnuksella, jolloin saadaan konejärjestelmän yhtenäinen ja linkitetynt kokonaismalli, johon älykäs virtuaalinen konelaboratorio pohjautuu [42]. Itse teknistä suunnitteluprosessia voidaan helpottaa suuresti semanttisen mallinnuksen keinoin ja automatisoida tietokoneelle esimerkiksi ihmiselle työläs suunnittelutiedon eheyden tarkastelu [36].

Tietokoneet ovat olleet pitkään luonnollinen apu tietojenkäsittelyssä. Pelkkä tiedon digitalisointi ei kuitenkaan yksin auta valjastamaan tietokoneita ratkomaan tietojenkäsittelyn ongelmia, vaan myös tiedon merkitys halutaan muuttaa formaaliksi tietokoneiden käsittelemään muotoon. Semanttisen webin standardit vaikuttavat lupaavilta ratkaisuilta, ja ne on havaittu hyödyllisiksi jo monella soveltavalla osalueella, kuten tuotetiedon mallintamisessa [9], lääketieteessä [12] ja ohjelmistotekniikassa [49]. Rakenteisen koneluettavan tiedon – avoimen datan – julkaisu on myös osa julkishallinnon strategiaa Suomessa; avoin data auttaa lopulta ihmisten arkipäiväistä elämää esimerkiksi erilaisten web-sovellusten muodossa [52].

Semanttisen webin tekniikoilla tarkoitetaan niitä T. Berners-Leen vuonna 2000 esittämä vision pohjalta kehitettyjä tekniikoita (kuvassa 4.1), jotka mahdollistavat WWW:n tehokkaamman hyödyntämisen tiedon jakeluvälineenä [8].

W3C haluaa kehittää tiedon julkaisua WWW-ympäristössä linkitettyjen dokumenttien tasolta linkitetyn tiedon tasolle [55]. Kuvan 4.1 visio esittää, että WWW:ssä olevalla tiedolla on sen yksilöivä URI-tunniste, tieto on digitaalisessa muodossa, sillä on rakenteinen merkkaus, semanttinen merkitys, se liittyy olemassa



Kuva 4.1: Semanttisen webin kerrosarkkitehtuuri. T. Berners-Leen vuonna 2000 esittämän vision [8] päivitetty versio [33].

oleviin tiedon merkitystä kuvaileviin ontologioihin ja siihen voidaan soveltaa päätelysääntöjä. Kuten kuka tahansa voi julkaista tietoa dokumentteina webissä, niin halutaan antaa ihmisille mahdollisuus linkittää dokumenttien sijaan tietoa itsessään, kuvailla sen merkitystä sekä lisätä päättelysäännöksiä tiedolle.

Seuraavissa luvuissa käsitellään semanttista tiedon mallinnusta koneluettavaksi RDF:n avulla, tiedon merkityksen eli ontologiatiedon kuvaamista RDFS- ja OWL-tekniologiolla, koneen tekemää päättelyä ja semanttisen metadatan kuvailua attribuutteina XHTML-dokumentteihin RDFa:n avulla.

Nämä teknologiat ovat ilmaiseksi kaikkien käytettävissä, ne ovat standardoituja ja teknisesti tuettuja. Tässä diplomityössä halutaan selvittää näiden teknologioiden käyttöä sellaisen tietomallin luomiseksi koneensuunnittelun tiedolle, että älykkäät sovellukset ovat yksinkertaisia toteuttaa.

4.1.1 RDF

Semanttinen web on nykyisen webin laajennus ja visio siitä, miten toteutetaan seuraavan sukupolven älykäs WWW-järjestelmä siten, että sen sisältämää tietoa paremmin hyödyntävät tietokoneet tarjoavat parempia sovelluksia käyttäjille. RDF

(englanniksi Resource Description Framework) on kuvauskieli ja mallinnustapa tiedolle tietokoneen ymmärtämässä muodossa, joka pyrkii ratkaisemaan osan vision asettamista ehdoista. RDF:ssä tieto on mallinnettu lausumina resursseista subjekti-predikaatti-objekti-kolmikkoina. RDF-kieli sarjallistetaan usein XML:ksi, mutta muitakin vaihtoehtoja on, kuten ihmiselle helppolukuisempi Notation 3 (niin kutsuttu N3). [11]

Esimerkiksi verkkosivusta voidaan kuvata täsmällistä koneluettavaa tietoa. Esimerkkinä ilmaistaan internet-hakukonesivuston tiedot.

Verkkosivulla, jonka identifioi sen domain-osoite 3g2upl4pq6kufc4m.onion,

- on nimi DuckDuckGo;
- on kuvaustieto sivuston sisällöstä;
- on suhde URI-osoitteeseen duckduckgo.com;
- on tietty aihepiiri
- ja se on tyypiltään hakukone.

Nämä tiedot halutaan ilmaista digitaalisesti, formaalisti, koneluettavasti ja sarjallistettuna XML:ksi. Ratkaisuna on kirjoittaa verkkosivun kuvailutiedot RDF:nä seuraavan esimerkin mukaisesti.

```

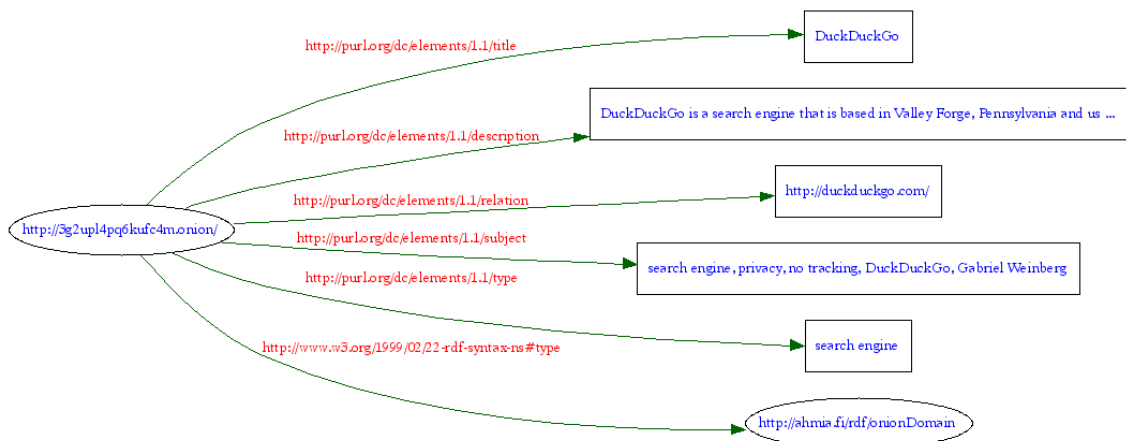
1 <rdf:Description rdf:about="http://3g2upl4pq6kufc4m.onion/">
2   <dc:title>DuckDuckGo</dc:title>
3   <dc:description>DuckDuckGo is a search engine that is based in Valley Forge,
      Pennsylvania and uses information from crowd-sourced sites (like Wikipedia)
      with the aim of augmenting traditional results and improving relevance.
      The search engine philosophy emphasizes privacy and does not record user
      information.</dc:description>
4   <dc:relation>http://duckduckgo.com/</dc:relation>
5   <dc:subject>search engine, privacy, no tracking, DuckDuckGo</dc:subject>
6   <dc:type>search engine</dc:type>
7   <rdf:type rdf:resource="http://ahmia.fi/rdf/onionDomain"/>
8 </rdf:Description>

```

Selkeytyksen vuoksi esimerkistä on jätetty XML-nimiavaruuksien esittelyt pois. XML-nimiavaruudet estävät varattujen sanojen yhteentörmäykset; tässä esimerkissä dc-lyhennettä vastaa Dublin Core -metatietosanastostandardin yksikäsitteinen URL-osoite. Yleisesti käytössä oleva Dublin Core esittelee esimerkissä käytetyn title-käsitteen siten, että tietokone osaa käsitellä tietoa sen mukaisesti, että elementin sisällä oleva merkkijono tarkoittaa semantiikaltaan otsikkoa. Vastaavasti voidaan koneellisesti käsitellä esimerkissä kuvatusta tiedosta lausumat ”3g2upl4pq6kufc4m.onion-on tyypiltään-hakukone”.

Koska RDF:ssä lausumat ovat kolmikkoina (subjekti-predikaatti-objekti), jossa subjektilla ja predikaatilla on omat URI-osoitteensa ja objekti on URI tai se on merkkijono, niin tiedosta syntyy tietorakenne, joka voidaan aina esittää graafina. Yhteisten URI-osoitteiden avulla erilliset graafit voidaan yhdistää keskenään yhteiseksi graafiksi.

RDF-tiedosto voidaan validoida samaan tapaan kuin verkkosivut W3C:n ylläpitämässä validaattorihjelmassa ¹. Samalla on mahdollista pyytää validaattoria esittämään tieto graafina. Kuvassa 4.2 on esitetty graafiksi visualisoituna tässä esimerkkinä käytetty tietomalli.



Kuva 4.2: W3C:n RDF-validaattorin tuottama graafivisualisointi semanttisesta mallista.

Kaikki nuolet kuvaavat predikaatteja. Ne lähtevät subjektista ja päättyvät objektiin. Tässä esimerkissä subjekteja on vain yksi ja jokaiseen objektiin päättyy vain yksi nuoli; samaan objektiin päättyisi useampia nuolia, jos se olisi esimerkiksi eri subjektia koskevan lausuman objektina.

4.1.2 Skeemat ja ontologiat

RDF:llä on mahdollista kuvata resurssien välistä tietoa lausumilla. Lausumien lisäksi halutaan tarjota jaettuja käsitteistöjä, joiden avulla tietoja kuvataan. Jaetun käsitteistön määrittelyminen RDF-kielellä ei onnistu, vaan tarvitaan laajennuksia kuvaamaan niin sanottua ontologiatietoa. Ontologiatiedolla tarkoitetaan tietyn aihepiirin käsitteiden määrittelyä ja kuvaamista formaalisti. [29]

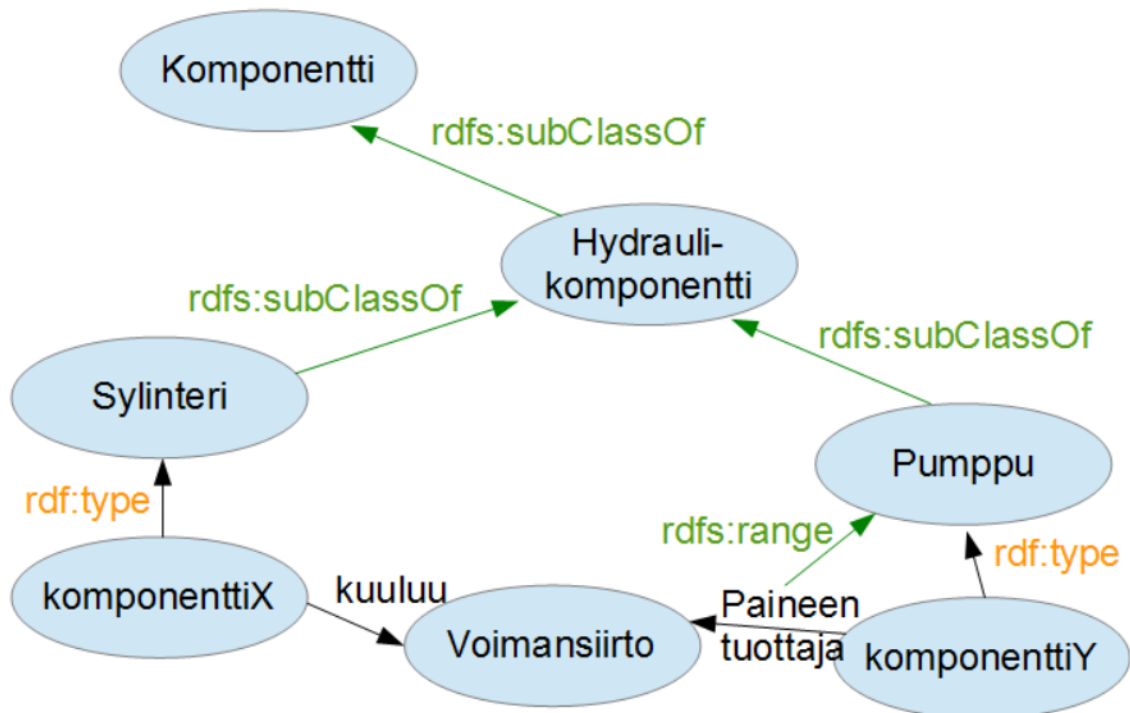
RDFS

RDF-skeema (RDFS) laajentaa RDF:n ilmaisuvoimaa tarjoamalla mahdollisuuden kuvata taksonomioita, luokkia ja ominaisuuksia [29]. Siinä, missä semanttisen webin mallinnuskieli RDF liittyy tiedon käsittelyyn graafina, niin RDFS liittyy tiedon

¹<http://www.w3.org/RDF/Validator/>

käsittelyyn joukkoina. RDFS:n avulla voidaan lausua, minkä tyyppiset instanssit liittyvät toisiinsa, ja miten ne liittyvät toisiinsa. [5, s. 91].

Tarkastellaan skeematietoa esimerkin avulla (kuvassa 4.3).



Kuva 4.3: RDF-lausumien graafi, jossa on RDF-skeematietoa on rdfs-alkuiset tiedot.

Tietomalliin on ilmaistu tietoja komponenttien luokittelusta, instanssien sylinterikomponentti X, pumppukomponentti Y ja voimansiirto välisistä suhteista ja paineen tuottamisen käsitteestä. Kuvassa olevat tiedot voidaan lukea suomeksi:

- Komponentti X kuuluu voimansiirtoon.
- Komponentti X on sylinteri.
- Komponentti Y tuottaa paineen voimansiirrolle.
- Painetta tuottavat komponentit ovat pumppuja.
- Sylinterit ovat hydraulikomponentteja.
- Pumput ovat hydraulikomponentteja.
- Hydraulikomponentit ovat komponenttien osajoukko.

RDF ja RDFS mahdollistavat näiden tietojen kirjoittamisen siten, että tietokone voi myös hyödyntää lausumien semanttista merkitystä päättelämällä tietoa tiedosta. Luokittelusta voidaan havaita kaikkien sylintereiden olevan hydraulikomponentteja

ja kaikkien hydraulikomponenttien komponentteja. Koska X on sylinteri, niin se on myös hydraulikomponentti ja komponentti. Tämän uuden tiedon tuottamista kutsutaan päättelyksi.

Päättelyn voi suorittaa niin sanottu *päättelijä* (päättelykone)², joka on sääntökieltä käyttävä tietokoneohjelma.

Kuva 4.3 näyttäisi päättelyn jälkeen sellaiselta, että jokaiseen instanssiin olisi täydennetty puuttuvat lausumat. Tällaisia ovat esimerkiksi lausumat ” X on komponentti.” ja ” Y on hydraulikomponentti.”.

OWL

RDFS tarjoaa hyvin rajalliset mahdollisuudet mallintaa tiedolle käsitteistöä. Sen sijaan OWL (Web Ontology Language) on hyvin ilmaisuvoimainen kuvaamaan tiedon semanttista merkitystä [5, s. 123]. OWL-kielestä on olemassa versiot 1 ja 2 sekä erilaisia suppeampia aliversiota. Näistä OWL 2 Full -kieli on ilmaisuvoimaisin. [53]

OWL mahdollistaa tehokkaan käsitteellisen mallinnuksen ja on yleisesti käytössä useissa eri käyttökohteissa. OWL:n käyttöä helpottavat useat eri muokkausohjelmat, sitä käyttävät ohjelmistorajapinnat ja päättelykoneet.

OWL:lla voidaan muun muassa esittää sellaisia (matematiikan termien kanssa analogisia) käsitteitä, kuten käänteisrelaatio, transitiivisuus, symmetrisyys, ekvivalentsuus ja funktionaalisuus.

Otetaan esimerkki OWL:lla lausuttavasta ontologiatiedosta, jota ei voida tuottaa RDFS:n avulla ja testataan samalla ontologian editointia ontologiaeditorilla.

Voidaan kirjoittaa RDF:n avulla, että ”Asukas asuu asunnossa” ja ”Asunto asuttaa asukkaan”, mutta RDF:n tai RDFS:n avulla ei voida kirjoittaa käänteisrelaatiösääntöä siitä, että nämä asumiseen liittyvät lausumat ilmaisevat samanlaista instanssien välistä suhdetta.

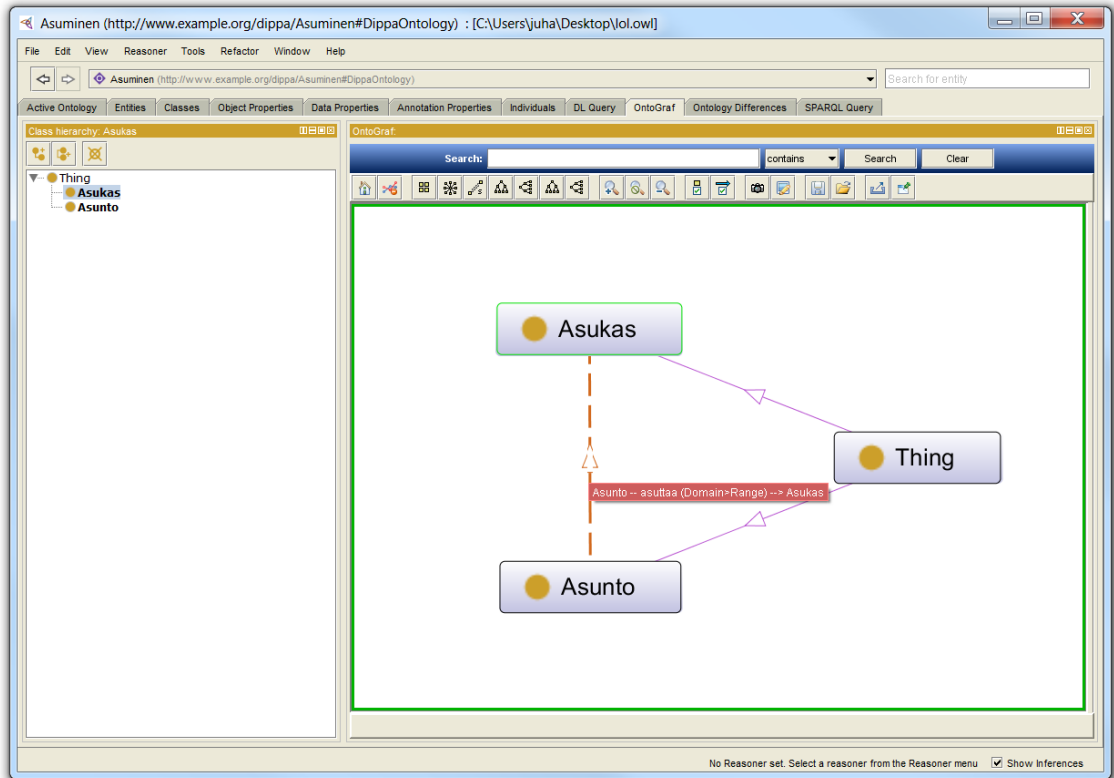
²Semanttisten sovellusten yhteydessä englannin kielellä esiintyvät synonyymit *semantic reasoner*, *reasoning engine* tai *reasoner*.

Kirjoitetaan ensin yksinkertainen OWL-esimerkkiontologia asukkaista ja asunnoista, sekä kuvataan asunnon ja asukkaan väliseksi ominaisuus ”asuttaa”.

```
1 <owl:Ontology rdf:about="http://www.example.org/dippa/Asuminen#DippaOntology">
2   <rdfs:label>Esimerkkiontologia.</rdfs:label>
3   <rdfs:comment>Asumisen ontologia.</rdfs:comment>
4 </owl:Ontology>
5
6 <owl:Class rdf:about="http://www.example.org/dippa/Asuminen#Asukas">
7   <rdfs:label>Asukas</rdfs:label>
8   <rdfs:comment>Asukas asuu jossain.</rdfs:comment>
9 </owl:Class>
10
11 <owl:Class rdf:about="http://www.example.org/dippa/Asuminen#Asunto">
12   <rdfs:label>Asunto</rdfs:label>
13   <rdfs:comment>Asunto on paikka, jossa asutaan.</rdfs:comment>
14 </owl:Class>
15
16 <owl:ObjectProperty rdf:ID="asuttaa">
17   <rdfs:domain rdf:resource="http://www.example.org/dippa/Asuminen#Asunto"/>
18   <rdfs:range rdf:resource="http://www.example.org/dippa/Asuminen#Asukas"/>
19 </owl:ObjectProperty>
```

Ontologia määrittelee luokat ”Asukas” ja ”Asunto” sekä niiden välisen ominaisuuden ”Asuttaa”. ”Asuttaa”-ominaisuus on rajattu siten, että se on predikaattina ”Asunto”-subjektin ja ”Asukas”-objektin välissä. Ontologiaan on lisäksi kuvattu ontologian metatietoa, kuten ontologian nimi ja tarkoitus, nimikekenttiä sekä kommenttikenttiä.

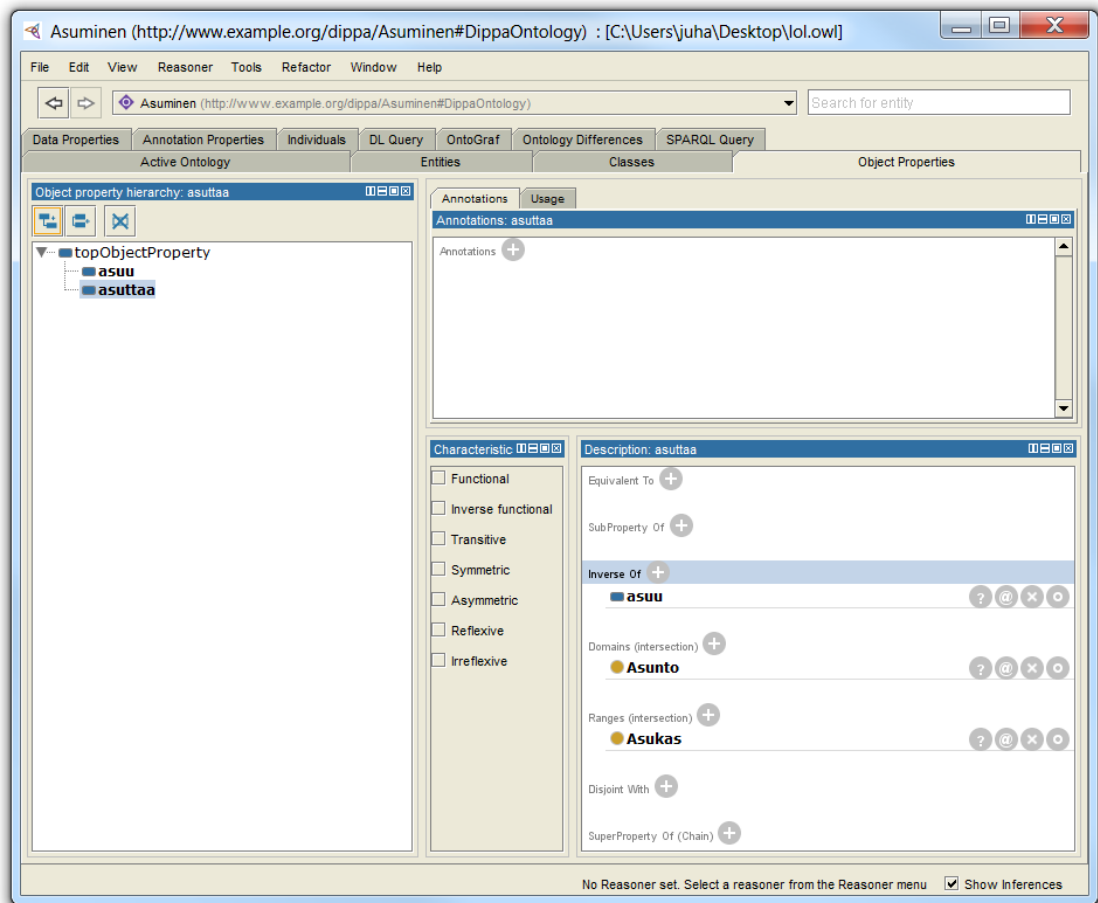
Lisätään ”Asuttaa” ominaisuudelle käänteisrelaatio ”Asuu” esimerkkiontologiaan. Ontologioita kannattaa muokata erityisesti ontologioiden hallintaan tarkoitetuilla ohjelmistoilla. Avataan esimerkkiontologia Protégé-nimisellä suosituilla ja ilmaisella ontologioiden hallintatyökalulla³. Editorilla voi paitsi muokata ontologiaa, myös tarkastella käsitteiden välisiä suhteita ontologiassa (kuvassa 4.4).



Kuva 4.4: Protégé-ohjelmiston käyttöliittymä. Ontologia on avattu tarkasteluun graafina.

³Protégé-ontologiaeditori, <http://protege.stanford.edu/> (14.07.2012)

Protégé-editorin käyttöliittymän kautta voidaan lisätä uusi ominaisuus ”asuu” ja merkitä sen olevan (owl:inverseOf) käänteisrelaatio ominaisuudesta asuttaa. Kuvassa 4.5 tarkastellaan ontologiassa olevia ominaisuuksia Protégén editorinäkymässä.



Kuva 4.5: Ontologiaan on lisätty uusi ominaisuus ”asuu” Protégén ominaisuuseditorilla ja merkitty se käänteisrelaatioksi ominaisuudesta ”Asuttaa”.

Nyt ontologian avulla on mahdollista päätellä, että jos ”AsuntoInstanssi1 asuttaa AsukasInstanssi1:n”, niin ”AsukasInstanssi1 asuu AsuntoInstanssi1:ssä” tai vastavasti päättely voi tapahtua toisinpäin.

Esimerkin mukaisesti voidaan rakentaa OWL:n ilmaisuvoimaa käyttäviä ontologioita RDF-malliselle tiedolle ja soveltaa päättelyä tietoon.

4.1.3 RDFa

Keskeinen semanttisen webin tavoite on saada WWW-julkaisu dokumenttikeskeisestä julkaisusta tietokeskeiseksi julkaisuksi, ja näin luoda älykkäämpiä sovelluksia loppukäyttäjien tarpeisiin. HTML-dokumentti ei ole helposti koneluettavissa elementtien käyttötavan vuoksi; edes otsikkoelementin sisällä ei välttämättä ole sitä seuraavaa leipätekstiä kuvaava otsikkoteksti. Toisaalta tiedon julkaiseminen erikseen ihmisten katseltavaksi HTML-dokumenttina ja siitä - esimerkiksi XSLT-muunnoksella - koneluettavaksi RDF:ksi tuo lisätyötä web-julkaisuun. W3C:llä on tähän ongelmaan ratkaisu, jonka käyttöönottokynnys on haluttu tehdä mahdollisimman matalaksi: yksinkertainen syntaksi, jolla voidaan rikastuttaa HTML-dokumenttiin RDF-merkintöjä. Nämä niin sanonut mikroformaattit ovat käytännössä lisäattribuutteja HTML-elementeille, jotka lisäävät dokumentin koneluettavuutta [5, s. 64].

W3C:n standardi RDFa (Rich Structured Data Markup for Web Documents) on mikroformaattistandardi HTML-sivujen rikastamiseksi metadata-attribuuteilla siten, että niistä on mahdollista hakea RDF-muotoista tietoa. Koska W3C on määritellyt yhteisen sanaston eli lisäattribuuttien nimet, niin kaikki RDFa:lla rikastetut WWW-sivut voidaan käsitellä samalla tavalla. [54] Sen sijaan, että olisi olemassa useita keskenään erilaisia tapoja merkitä metatiedot RDF-tiedon tuottamista varten, niin samojen attribuuttien käyttö mahdollistaa yhden XSLT-muunnoksen käytön kaikille sivuille [5, s. 64].

RDFa palvelee semanttisen webin tavoitetta parempien sovellusten luomiseksi loppukäyttäjille. Esimerkiksi hakukonejätti Google osaa käsitellä RDFa-merkinnät ja hyödyntää niitä hakutuloksien löytämisessä ja esittämisessä. Google ohjeistaa ruokareseptien julkaisuun RDFa-metamerkinnoilla rikastettuna ja näyttää suoraan hakutulossivulla reseptin keskeisimpiä tietoja, kuten ruuanlaittoon kuuluvan kokonaisajan⁴.

⁴Googlen web-sivujen tuottamisohje,
<http://support.google.com/webmasters/bin/answer.py?hlfi&answer=173379> (16.07.2012)

Ruokareseptin HTML-sivuun voidaan kuvata RDFa-merkinnöillä muun muassa kalorimäärä, reseptin saamat arvostelut, annoksien lukumäärä, yhden annoksen kalorimäärä, rasvan määrä annoksessa ja erikseen käytettävien valmistusainemäärien määrät. Seuraavassa esimerkissä on merkitty yhden annoksen ominaisuuksista tarjoiluannoksen koko, yhden annoksen kalorimäärä ja sen sisältämä rasva RDFa-standardin mukaisesti.

```
1 <span typeof="v:Nutrition">
2   Annoksen koko:
3   <span property="v:servingSize">1 keskikokoinen pala</span>
4   Kaloreita annoksessa:
5   <span property="v:calories">250</span>
6   Rasvaa annoksessa:
7   <span property="v:fat">12 g</span>
8 </span>
```

Metatietojen lisäys voidaan automatisoida julkaisujärjestelmän osaksi ja ne eivät vaikuta siihen, miten selain näyttää sivuston. Sen sijaan ne parantavat huomattavasti dokumentin koneluettavuutta ja lopulta palvelevat paremmin julkaistua tietoa hyödyntävää käyttäjää.

4.2 Web-pohjaiset sovellukset

4.2.1 Käyttöliittymä selaimessa

Jos VML halutaan toteuttaa webin tekniikoilla toimimaan selaimessa, niin käytännössä tarvitaan käyttöön aivan uusimpia nykyaikaisia selainteknologioita. Käydään seuraavaksi läpi haastavuudeltaan keskeisimpiä web-tekniikoita VML:n kannalta.

World Wide Web ja selainohjelmat ovat mahdollistaneet monien palveluiden siirtymisen fyysisiltä toimipaikoilta (esimerkiksi pankit ja kaupat) tai ulkopuolisista ohjelmistoista (esimerkiksi sähköpostiohjelmat ja pikaviestimet) selaimissa suoritettaviksi. Selaimen käyttö on muuttunut HTML-dokumenttien katseluohjelmasta ohjelmointia tukevaksi alustaksi ja siihen on tulossa entistä enemmän ohjelmointia tukevia ominaisuuksia [23]. Myös ohjelmistoteollisuudessa on tapahtumassa paradigman muutos: hyvin monet erilaiset ohjelmistot voidaan toteuttaa toimimaan selaimessa ja ohjelmistoteollisuus suhtautuu selaimen entistä vakavammin ohjelmointialustana [30]. Selaimessa toimivat ohjelmistot eivät vaadi asennusta ja ne eivät ole riippuvaisia tietystä käyttöjärjestelmästä.

3D-grafiikka selaimessa

Selainta voidaan käyttää alustana myös 3D-visualisointia vaativien ohjelmien, kuten virtuaalikonelaboratorion, ohjelmistokehityksessä. Tämä muutos on seurausta

kahdesta uudesta merkittävästä selainten toimintaa rikastuttavasta standardista: HTML5-standardista (seuraava pääversio HTML-kielestä) ja WebGL-standardista (Web-based Graphics Language) [19].

WebGL on 3D:n esittämiseen selaimessa tarkoitettu rajapinta [19]. WebGL käyttää näytönohjaimen OpenGL-kykyistä ajuria. WebGL:n toiminta pohjautuu vahvasti OpenGL-standardiin (Open Graphics Library) [19], joka on suorituskykyinen yksinkertaisia funktiokäskyjä tarjoava grafiikkarajapinta, jonka piirto- ja käskyt suoritetaan näytönohjaimella. Selaimissa ei ole ollut ennen WebGL-standardia tarjolla suorituskykyistä 3D-grafiikan tukea. [30]

Yksinkertaisista funktiokäskyistä koostuva grafiikkarajapinta WebGL tarjoaa suoraan mahdollisuuden ohjelmoida 3D-grafiikkaa selaimen, mutta sen käyttäminen on hyvin työlästä. Keskeiset ominaisuudet, kuten kappaleiden pyörittely, liikuttelu ja 3D-maailmassa liikkuminen pitää toteuttaa itse ja jokainen kappale pitää piirtää WebGL-tason rakenteiden avulla. Selaimessa tapahtuvan 3D-ohjelmoinnin helpottamiseksi on toteutettu erilaisia WebGL-rajapintaa käyttäviä abstraktiotasoa nostavia grafiikkakirjastoja, jotka tarjoavat 3D-piirron kannalta keskeisiä toimintoja valmiina [30].

Eräs tällainen kirjasto on kokeellinen vapaan lähdekoodin X3DOM-JavaScript-kirjasto. X3DOM pyrkii ratkaisemaan deklarattiivisen 3D-sisällön tuonnin HTML5-dokumentin osaksi; X3DOM lisää 3D-sisältöä kuvaavaa X3D-formaattia [14] osaksi dokumenttia ja selaimen dokumentin esitys- ja ohjelmointirajapintaa eli DOM-puuta. [57] 3D-sisällön käsittely tapahtuu JavaScript-ohjelmointikielen avulla ja se ei vaadi abstraktiotason alla toimivan WebGL-piirto- ja ohjelmointirajapinnan toiminnan ymmärtämistä. Tämä tarkoittaa esimerkiksi sitä, että kirjasto toteuttaa ohjelmoijan käskyn ”piirrä pallo”, jolloin ohjelmoijan ei tarvitse tuntea pallon piirtoalgoritmia.

Ohjelmointi ja suorituskyky

Selaimen käyttö on muuttunut HTML-dokumenttien katseluohjelmasta ohjelmointia tukevaksi alustaksi. Selainohjelmistoihin ja verkkosivuihin on jo pitkään tuotettu dynaamista toiminnallisuutta JavaScript-ohjelmointikielen avulla [25]. JavaScript-ohjelmointikieltä on laajennettu, selaimen tarjoamia rajapintoja lisätty ja HTML-kielen ominaisuudet kehittyvät koko ajan [23]. Tämä tekee selaimen houkuttelevaksi ohjelmointialustaksi myös VML:n toteuttamiseen.

Selaimen tukemin teknologian voidaan toteuttaa 3D-grafiikkaa ja simulointia vaativia ohjelmistoja. JavaScript on riittävän suorituskykyistä jopa peliohjelmoinnin tarpeisiin. [31]

Lisäksi ilmaisia käyttöliittymäkomponenttikirjastoja on laajalti saatavilla JavaScript-ohjelmointia varten.

Web-ohjelmoinnin haasteena on edelleen se, että ohjelmistoa on vaikea testata toi-

mivaksi erilaisille selainohjelmille. Jos halutaan tukea vanhentuneita selainohjelmistoja, niin ohjelmointityön määrä moninkertaistuu. Tämän lisäksi JavaScript on (nimensä mukaisesti) suunniteltu ”scriptikieleksi”, joka näkyy ylläpidettävyysongelmina kun ohjelmiston koko kasvaa. Nämä haasteet korostuvat isoissa web-ohjelmistoissa ehkä siksi, että web-ohjelmoinnissa vaikuttavat tietynlaiset kansanperinteet ja ei ole vakiintunut tietynlaista web-ohjelmistokehityksen ohjelmistotuotantoa. [30]

4.3 Datalähtöisyys ohjelmistokehityksessä

4.3.1 Datalähtöisyys paradigmana

Datalähtöisessä ohjelmoinnissa data itsessään ohjaa ohjelman toimintaa. Ohjelman logiikka määrittää vain geneerisen tapahtumakäsittelyn tiedolle. Esimerkiksi yksinkertainen XML-tietomallin visualisointi SVG-kuvaksi prosessoimalla se XSL-muunnoksen avulla putkilinjan läpi. Tällöin muunnosprosessin logiikka tuottaa eri lähtödatasta erilaisen SVG-kuvan. [32]

Datalähtöisyyden perusajatuksena on kirjoittaa ohjelmakoodia, jossa ohjelman käyttämä data (esim. lukuarvo) ei ole kovakoodattuna ohjelman koodissa. Sitä vastoin ohjelmistoon on koodattu abstraktien tietotyyppien käsittely siten, että ohjelman toiminta tapahtuu sen käyttämän datan avulla. Esimerkiksi ohjelma voi olla ohjelmoitu toteuttamaan toimintoja äänimerkki, animaatio ja tekstin näyttäminen datalähtöisesti. Tällöin voidaan lukea ohjelmaan data äänimerkki1, animaatio1, animaatio2 ja äänimerkki2, jolloin ohjelma suorittaa datan mukaisesti toiminnot.

4.3.2 Semanttisen tietomallin hallinta ja jakelu

SPARQL ja hakurajapinnat

SPARQL (SPARQL Protocol and RDF Query Language) on RDF-tietomallien kyselykieli, jolla voidaan formalisoida kysymyksiä tietomallille [40]. Kieli muistuttaa relaatiotietokantojen SQL-kyselykieltä. Vastaavasti, kuten SQL-kielellä, kielen uusimmalla versiolla SPARQL 1.1 voi tehdä myös päivityksiä RDF-tietomalliin [21]. W3C valmistelee myös standardia REST-arkkitehtuurin avulla RDF-tietomallin hallitsemiseksi [38].

4.3.3 Webin teknologioita hyödyntäviä esimerkkijärjestelmiä

DITA

DITA (Darwin Information Typing Architecture) on XML-pohjainen arkkitehtuuri dokumenttien julkaisemista varten. DITA:lla lopullinen julkaisu tuotetaan dataläh-

töisesti eri kirjoittajien tekstilähteistä ja sillä on selkeä tapa hallita eri informaatio-kokonaisuuksia [41].

DITA-arkkitehtuurin keskeiset ominaisuudet ovat:

- XML-pohjaisen rakenteisen tiedon luonti, hallinta ja julkaisu.
- Kirjoittajien yhteistyö, työprosessi ja työnjako.
- Tiedon jakelu ja uudelleenkäytettävyys.
- Kirjoitus- ja tuotantokustannusten laskeminen.
- Avoimet standardit.

Alunperin IBM:n kehittämä DITA on käytössä useissa suurissa projekteissa, joissa tarvitaan tehokasta rakenteista informaation hallintaa. Arkkitehtuuritoteutuksia löytyy kaupallisista ja avoimen lähdekoodin ohjelmistoista. [41]

Callimachus

Callimachus on sovelluskehys datalähtöisten web-sovellusten ohjelmointiin, joka hyödyntää semanttisen webin teknologioita. Callimachus tähtää W3C:n standardien helppoon käyttöönottoon ja semanttisen webin sovellusten luontiin. [24]

Callimachus toimii datan hallintajärjestelmänä, tietovarastona ja sovelluskehiksenä linkitetylle datalle. Hallintajärjestelmänä Callimachus mahdollistaa tiedon katselun, luomisen ja muokkaamisen sovelluskehittäjän määrittelemien näkymien avulla wikityyppisessä editointiympäristössä. Dataa voi julkaista sovelluksina, ihmisille helposti luettavassa olevina sivuina tai sitten tietokoneita varten REST- ja SPARQL-rajapintojen kautta. [24]

Ohjelmisto muistuttaa sisällönhallintajärjestelmää, mutta suurin ero on sen käyttämät semanttisen webin teknologiat ja rakenteisen datan suuri hyödyntäminen. Yleensä sisällönhallintajärjestelmät käsittelevät dataa tekstimuodossa, mutta eivät esimerkiksi hyödynnä RDF-tietoja ja ontologioita. [24]

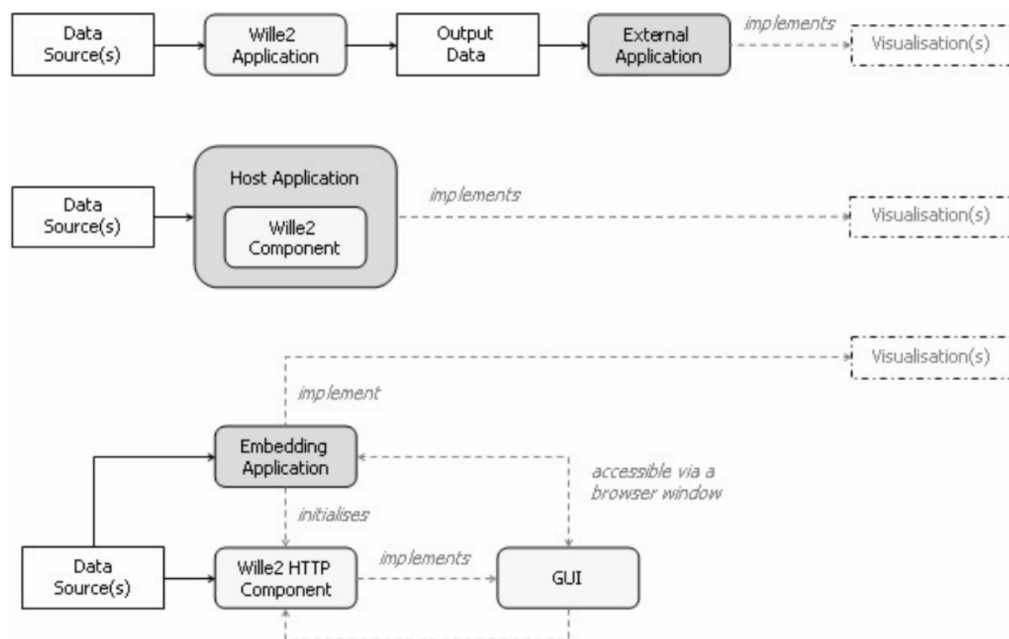
Toisaalta Callimachus sisältää paljon tietovarasto-ohjelmiston piirteitä eli sen avulla on mahdollista yhdistää erilaisista tietokannoista dataa yhdeksi hyvin analysointia, raporttien luomista ja kyselyjä palvelevaksi järjestelmäksi. Datasta voidaan tuottaa myös erilaisia analyttisiä tilastollisia yhdistelmänäkymiä. [24]

Callimachus käyttää RDFa-merkkausta myös kyselykielenä sivupohjissaan, jolloin sovellukseen saa HTML-elementtien sisällön Callimachuseseen lisätyistä RDF-tiedoista [6]. RDFa:n käyttäminen kyselykielenä on ohjelmistojen kehittäjien oma idea. Sivupohjaan voidaan siis kirjoittaa, että kun sivupohjaa kutsutaan tietyllä RDF-mallissa olevan instanssin URL-osoiteparametrilla, niin sivu luodaan instanssin tietojen mukaisesti.

Wille

Wille on sovelluskehys datalähtöisten visualisointiohjelmistojen tekemiseen. Se sisältää HTTP-palvelimen ja mahdollistaa sovelluksen käytön selaimessa. Willen putkilinja määrittelee mahdollisesti useista lähteistä tulevan datan esikäsittelyn ja julkaisun. [45]

Willen datalähtöinen toteutustapa näkyy siten, että Wille toteuttaa prosessin datan käsittelylle (kuvassa 4.6). Itse data luetaan sisään toimintalogiikalle, jonka tuottamaa ulostulua voidaan hyödyntää. [35]



Kuva 4.6: Willen käyttömallit visualisointisovellusten luomiseksi. Wille toteuttaa datalähtöisen prosessin, jossa tietomalli ja sovelluslogiikka ovat erikseen.

Kuten DITA-arkkitehtuurissa, myös Willessä on yksinkertaista tunnistaa eri datan hyödyntämisen vaiheissa eri roolit. Willessä on datan tuottaja (Author), kehittäjä (Developer), suunnittelija (Designer) ja loppukäyttäjä (End-User). He voivat sopia yhteiset rajapinnat toistensa välissä ja toteuttaa itsenäisesti oman työnsä. [35]

Datalähtöisyys ja sen periaatteiden mukaisesti toimivat järjestelmät pyrkivät vähentämään ihmisten työmäärää, jota tarvitaan uuden tiedon yhdistelyssä ja käsittelyssä.

5. TAPAUSTUTKIMUS ASIAANTUNTIJAHAASTATTELULLA

Diplomityössä selvitetään virtuaalisen konelaboratorion ominaispiirteet yhteissuunnittelun tukityövälineenä sekä valitaan löydetyistä keskeisistä ominaisuuksista muutamia esimerkkitoteutusta varten ohjelmistoprototyypiin. Käsitys yhteissuunnittelun tukityövälineenä toimivasta VML:stä auttaa koko Semogen-hanketta ymmärtämään yhteissuunnittelun ongelmia, suunnittelemaan tarvittavaa tietomallia ja luo samalla pohjan ohjelmistoprojektille, jossa toteutetaan tutkimusprototyyppi VML-järjestelmästä.

Selvityksen taustalla olevat tutkimuskysymykset ovat:

- **1. Millainen on hyvä mekatronista yhteissuunnittelua tukeva VML?**
 - Mitä hyötyä VML:llä tavoitellaan?
 - Mitkä ovat VML:n suunnittelua tukevat käyttötapaukset?
 - Miten VML liittyy osaksi suunnitteluprosessia?

5.1 VML:n ominaisuuksien kartoitus

Ennen kuin Semogen 2 -hankkeessa alettiin toteuttamaan käyttötapauskohtaisia demottavia VML-näkymiä, tehtiin kartoitus siitä, millainen suunnittelua tukeva virtuaalinen konelaboratorio oikeastaan kuuluisi olla. Vaatimusten ja käyttötapauksien selvitys on keskeinen peruslähtökohta jokaiselle ohjelmistoprojektille [20]. Vaikka demottavat VML-näkymät ohjelmoidaan pääasiassa tutkimusryhmän tutkimuskäyttöön eikä niinkään tuotteeksi, niin silti oikeiden tunnistettujen käyttötapauksien toteuttaminen, sekä ylipäätään vaatimusten listaaminen antaa raamit ja konkretiaa Semogenissa tehtävälle tutkimukselle. Hankkeessa VML:n ominaisuuksien kehittäminen on osa iteratiivisen Top-Down suunnitteluprosessin tukemista.

VML:n vaatimuksien kartoittamiseksi käytetään tutkimusmenetelmänä kyselylomaketta ja sen kvantifointia. Kyselyn toteutus perustuu kvalitatiivisten tutkimusmenetelmien ja kvantifoinnin metodeihin [50]. Kyselylomakkeella kerätään palautetta ehdotettuihin mahdollisiin käyttötapauksiin sekä uusia ideoita ja käyttötapauksia. Saadut vastaukset litteroidaan ja kvantifoidaan. Tässä litteroinnilla tarkoitetaan vastaajien tuottamien vastausten kirjoittamista puhtaaksi sähköiseen muotoon. Työpajoissa pohjustetaan keskustelua VML:stä yleisinformaatiota antavalla

esityksellä ja samalla rajataan keskustelua VML:stä koneensuunnittelun alueelle.

Selvityksen avulla saadaan lopputulokseksi

- ymmärrystä VML:stä yhteissuunnittelun tukena;
- vaatimuksia VML:n tarvitsemalle suunnittelutiedolle;
- käsitteitä sekä niiden välisiä suhteita VML:ssä ja
- käsitys yhteissuunnittelun ongelmista, joita VML:n avulla voidaan helpottaa.

Käyttötapausten kvantifioinnissa käytetään erilaisia viitekehysjä, joihin vastauksissa saatuja käyttötappauksia sijoitellaan. Näin saadaan analysoitua paremmin sitä, millaisia ehdotetut käyttötappaukset ovat.

Käyttötappauksista valitaan keskeisimmät demosovelluksen kehittämisen pohjaksi ja pyritään toteuttamaan niitä vastaava toiminnallisuus suunnittelun tueksi data-lähtöisesti semanttisen mallinnuksen avulla.

5.2 Kyselyn tulokset

Kyselyssä saadut käyttötappausideat litteroitiin ja niistä haettiin erilaisten viitekehysten avulla lukuarvoista dataa eli käyttötappauksille suoritettiin kvantifointi. Yhteensä käyttötappausideoita saatiin 15 kappaletta Semogen-hankkeen teollisuuskumppaneilta. Vastajia oli yhteensä 18. Eniten vastauksia saatiin suunnittelujärjestelmiä kehittävältä Vertex Systemsiltä ja kallonporauslaitteiden tuottamiseen erikoistuneelta Sandvikilta. Vastajat ovat kaikki tutustuneet Semogen-hankkeeseen ja tuntevat työnsä puolesta koneensuunnitteluprosessin hyvin.

Kysely esiteltiin Semogen-hankkeen työpaajoissa yrityskumppaneille ja jätettiin kirjallisena vastajille, jotka palauttivat vastauksensa seuraavan parin viikon kuluessa.

Tuloksen toivotaan antavaan suuntaa siitä, mikä on suunnittelijoiden nykykäsitys suunnitteluprosessin tarpeista. On kuitenkin huomattava, että vastajilla ei ole käytössä tämän diplomityön kaltaista johdantoa yhteissuunnitteluun ja yhteisen työtilan visiointiin. He myös omaksuivat eri suunnittelualojen rooleja ideoidessaan parempaa suunnittelutyötä. Vastaukset heijastavatkin vastajien tietyn hetkistä käsitystä siitä, miten suunnittelutyötä pitäisi parantaa.

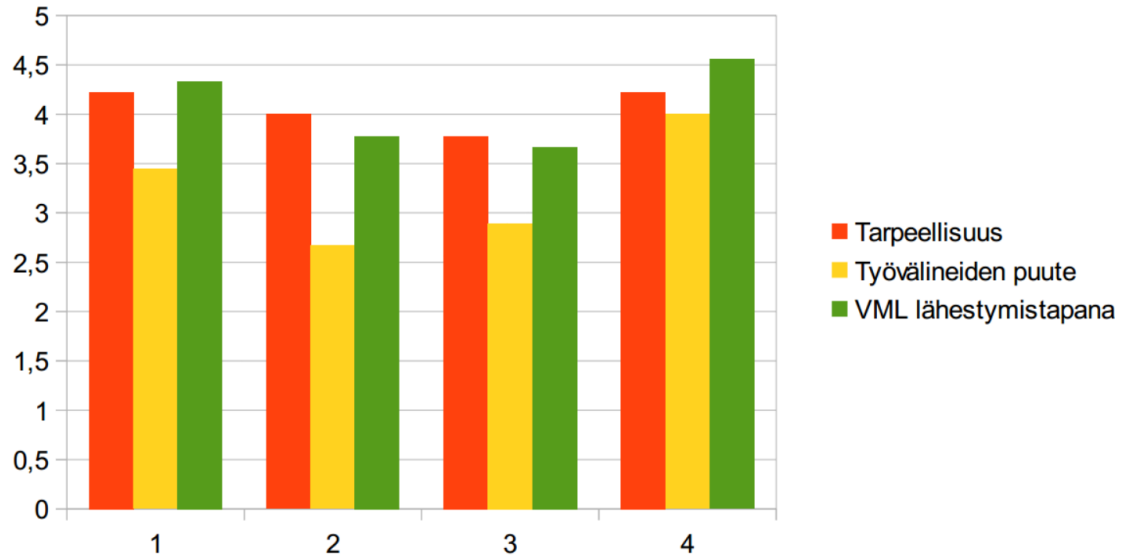
Vapaan käyttötappausideoinnin lisäksi Semogen 1- ja Metviro-hankkeen perusteella laadituista valmiina kyselyssä esitetyistä neljästä käyttötappauksesta, jotka ovat

1. toimintojen tarkastelu,
2. suunnittelijoiden välinen kommunikointi,

3. suunnitteluratkaisujen vertailu ja

4. virtuaaliprototyyppi,

kerättiin numeerista ja sanallista arviointia. Tulokset koostettiin kuvaajaksi, joka on kuvassa 5.1. Asteikko on 1-5. Mitä korkeampi lukuarvo, sitä tarpeellisempi, työvälineiltään puutteellisempi ja VML-ratkaisuna hyödyllisempi käyttötapaus.



Kuva 5.1: Esitettyjen käyttötapausten tarpeellisuus, havaittu työvälineiden puute nykyään ja VML:n järkevyyden lähestymistapana kyselyjen perusteella asteikolla 1-5. Mitä korkeammat palkit, sitä tärkeämpi käyttötapaus on.

Kuvasta 5.1 havaitaan, että käyttötapaus 4 (virtuaaliprototyyppi) nousee tarpeellisimmaksi, siihen on vähintään työvälineitä olemassa nyt, ja VML-ohjelmistoa pidetään oikeana lähestymistapana.

Samassa kyselyssä vastaajat antoivat lopuksi tärkeysjärjestyksen näille käyttötapauksille; kyselyn lopussa pyydettiin laittamaan vielä nämä neljä käyttötapausta tärkeysjärjestykseen. Lopputuloksena saatiin tärkeysjärjestys keskiarvojen mukaan 1, 2, 4 ja 3. Tärkeimmäksi käyttötapaukseksi valittiin eniten toimintojen tarkastelua (1) sekä suunnittelijoiden välistä kommunikointia (2), joita molempia valittiin yhtä monta kertaa tärkeimmäksi. Tämä vahvistaa entisestään käsitystä siitä, että suunnittelussa on tarve auttaa eri suunnittelun osa-alueita parempaan yhteistoimintaan kokonaisen toimivan konejärjestelmän tuottamiseksi.

Laadullisen kyselyaineiston kvantifointi tehtiin VML:n M1-tekniikan ominaisuuksien (taulukko 2.1), vaadittavan suunnittelutiedon konkretian ja yksityiskohdaisuuden (kuva 3.3) ja tietokoneavusteisen yhteisöllisen työn (kuva 3.1) antamien viitekehysten avulla.

Kvantifointimenetelmä toimi siten, että ensiksi esitetyn käyttötapausten vaatimat M1-tekniikan ominaisuudet merkittiin taulukkoon (kuvassa 5.2). Toiseksi

tarkasteltiin käyttötapauksen vaatimaa koneensuunnitteluinformaatiota konkreettisesti ja tarkkuusasteikoilla (kuvassa 5.3). Lopuksi tarkasteltiin, miten käyttötapaus sijoittuu CSCW-järjestelmien konseptien luokittelusteikolle ajassa ja paikassa (kuvassa 5.4).

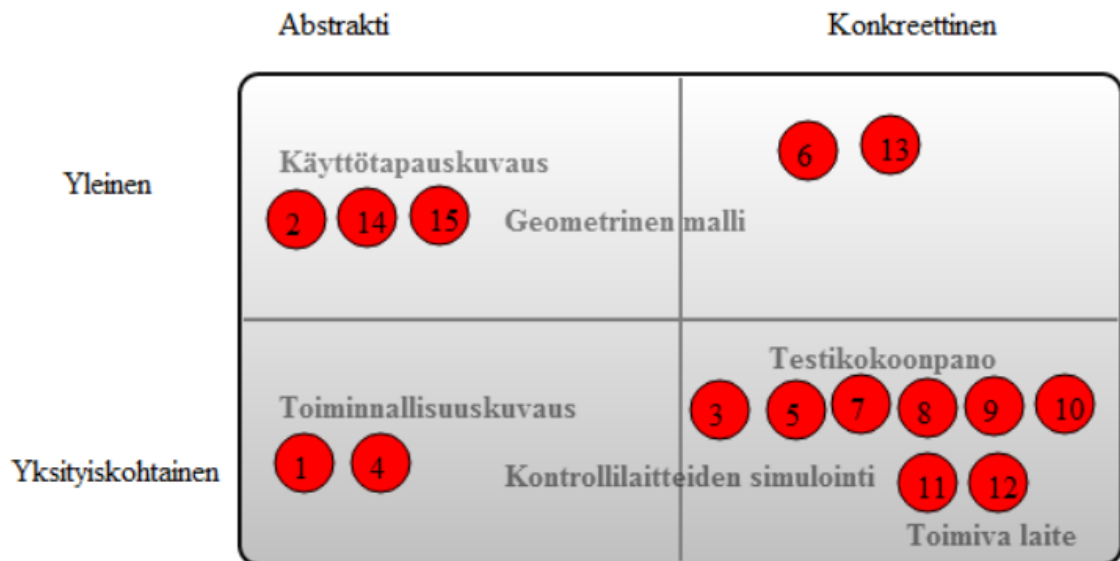
Kuvassa 5.2 on tulokseksi saadut käyttötapauksen ominaisuustarpeet Semogen 1-hankkeessa ja Metvirossa tunnistetuista ominaisuuksista. Viimeisessä sarakkeessa on lukuarvo, joka kertoo, kuinka usein ominaisuutta tarvitaan näitä käyttötapauksia toteutettaessa.

Käyttötapaus kyselyistä (nro.).	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
1. Konejärjestelmän muutosten tuki.	■	■		■		■	■						■			6
2. Dynaaminen reaaliaikaisimulaatio.							■	■		■	■	■			■	6
3. Selainpohjainen käyttöliittymä.																0
4. Semanttinen haku.			■	■	■	■							■	■		6
5. Muokattavat skeemat.	■	■		■		■	■						■			6
6. Visualisoinnit laitteesta.	■	■						■		■	■	■				7
7. Yhteistoiminnalliset näkymät.	■	■	■			■	■	■		■	■				■	9
8. Mittaukset.							■	■		■		■				4
9. Ohjauskontrollit.	■						■	■		■	■					6
10. Simulaation hallinta.							■	■				■			■	4

Kuva 5.2: Ehdotetut käyttötapaukset (1-15) ja niiden vaatimat Metvirossa ja Semogen1:ssä tunnistetut VML:n ominaisuudet. Esimerkiksi kyselyssä saatu käyttötapausidea numero 11 vaatii dynaamisen reaaliaikaisimulaation, visualisoinnit laitteesta, yhteistoiminnalliset näkymät sekä ohjauskontrollit.

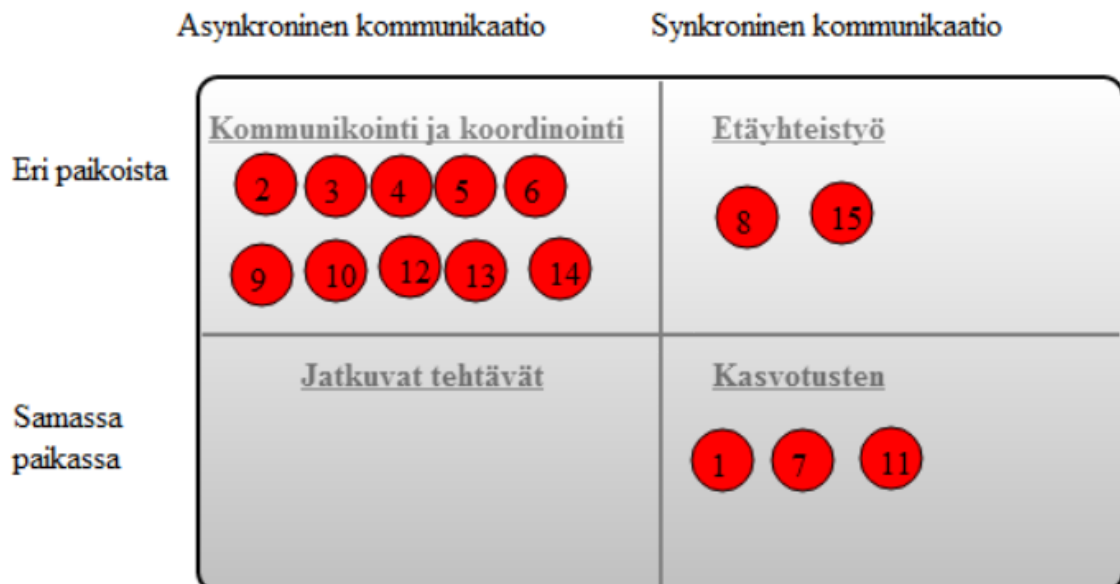
Näistä VML:n ominaisuuksista visualisoinnit laitteesta ja yhteistoiminnalliset näkymät laitteeseen ovat tarpeellisimmat ominaisuudet. Lisäksi vaikuttaa siltä, että selainpohjainen käyttöliittymä ei ole sinänsä vaadittava ominaisuus: se, toisin kuin muut ominaisuudet tässä listassa, ottaa kantaa toteutusteknologiaan. Hajautettavuus ja näkymän jakelu sopisi paremmin tämän ominaisuuden tilalle, jotta se olisi vaatimustarkkuudeltaan muiden listan ominaisuuksien mukainen.

Kuvassa 5.3 on käyttötapauksen tiedontarpeet suunnittelun malleilta. Kuvasta selviää käyttötapauksista, kuinka konkreettista ja tarkkaa suunnittelutietoa ne vaativat.



Kuva 5.3: Käyttötapauksen tiedontarpeet. Numeroitu pallo on kyselyssä saatu käyttötapausidea. Kuvasta havaitaan, kuinka konkreettista ja tarkkaa suunnittelutietoa tietty käyttötapaus vaatii.

Kuvassa 5.4 on esitetty tulos käyttötapauksen sijoittumisesta tietokoneavusteisen yhteisöllisen työn luokittelukentälle.



Kuva 5.4: Tietokoneavusteisen yhteisöllisen työn ohjelmistojen luokittelukehys, jota on käytetty käyttötapauksen luokitteluun. Numeroitu pallo on kyselyssä saatu käyttötapausidea.

Esimerkiksi kyselyssä on esitetty käyttötapausidea koneen käyttöliittymän tarkastelemiseksi, joka on litteroinnissa numero 11. Käyttöliittymän tarkastelun idea on, että suunnittelu, tuotepäällikkö ja myyjä voivat ajaa virtuaalista konetta ja testata näin käyttöliittymän sekä koneen toimivuutta ja esitellä laitetta.

Ensiksi katsotaan, mitkä kuvan 5.2 VML:n esitetyistä ominaisuuksista tarvitaan käyttötapausten 11 toiminnallisuuden toteuttamiseksi. Ominaisuuksiksi tarvitaan selkeästi

- 2. dynaamisen reaaliaikasisimulaation tuki,
- 6. visualisoinnit laitteesta,
- 7. yhteistoiminnalliset näkymät
- ja 9. ohjauskontrollit.

Toiseksi katsotaan, kuinka tarkkaa ja konkreettista suunnittelutietoa käyttötapaus vaatii. Jotta järjestelmää voidaan ajaa koneen käyttöliittymän avulla, niin koneen on oltava hyvin konkreettinen. Toisaalta laitteessa voi silloin olla vielä luonnosvaiheessa olevia alijärjestelmiä. Näistä syistä käyttötapaus 11 on sijoitettu hyvin konkreettiseksi (oikeaan reunaan) ja hieman yli puolen välin asteikolla luonnoksesta täsmälliseen tarkkaan toteutustapaan kuvassa 5.3.

Kolmas vaihe analyysissä on tarkastella, miten esitetty käyttötapaus sijoittuu konseptina ajassa ja paikassa tietokoneavusteiseksi yhteisöllisen työn välineeksi. Käyttöliittymien testaus tapahtuu yleensä ihmisten kesken yhteisessä samassa paikassa. Samoin esimerkiksi markkinointiin liittyvät esittelyt ja asiakkaalle laitteen käytön sekä toimintojen esittely tapahtuu synkronisesti samassa huoneessa. Tästä syystä käyttötapaus 11 sijoittui synkronisen kommunikaation ja samassa paikassa työskentelyn alueelle kuvassa 5.4.

5.2.1 Johtopäätökset kyselyn tuloksista

Analysoitujen kyselyissä ideoitujen käyttötapausten perusteella on mahdollista tehdä suuntaa antavia johtopäätöksiä siitä, millainen on VML suunnittelun tukena, mitä tietoja se vaatii, ja mitkä VML:n yleisistä ominaisuuksista ovat keskeisiä suunnittelun tukena toimivalle VML-ohjelmistolle.

Keskeisimmät johtopäätökset ovat:

- Suunnittelussa on tarve VML:lle: nykyiset ratkaisut koetaan riittämättömiksi.
- VML:n keskeisin tarkoitus on kommunikoinnin ja projektin koordinoimisen avustaminen.

- Suurin osa kyselyissä ideoiduista käyttötapauksista vaatii konkreettista ja tarkkaa suunnittelutietoa, joka vastaa todellista konetta.
- Erityisesti vaaditaan yhteistoiminnallisia näkymiä ja suunnitteluaineiston onnistunutta visualisointia.
- Suurin ongelma suunnittelussa, jota tulisi ratkoa VML-ohjelmistolla, on tiedon siirtämisessä ihmiseltä toiselle.
- Eri työtehtävissä olevat ihmiset haluavat nähdä toimialojaan yhdistäviä näkymiä.
- Mittausten tekeminen ja simulaation hallinta (hidastus tai toisto) on toissijaista.
- Noin puolet käyttötapauksista edellyttää dynaamista reaaliaikasilmoitusta tai ohjauskontrolleja laitteeseen.
- Toisaalta noin puolet käyttötapauksista ovat mahdollisia toteuttaa VML:ään ilman ohjattavaa liikkuvaa laitetta eli myös ilman simulointia.
- Useat käyttötapaukset vaativat sellaisen tiedon tuottamista, jota ei tällä hetkellä kirjata ylös.

Kysely antaa rajausta ja käsitystä siitä, millainen on VML suunnittelun tukena. Toisaalta kysely tuottaa menetelmänä tuloksiin mahdollisesti virheitä. Kyselyn rakenne ja johdanto konelaboratorioon saattaa johdatella vastaamaan tietyllä tavalla ja tuomaan esiin vain tiettyjä ideoita. Lisäksi vastaajia oli rajattu joukko tiettyistä yrityksistä ja ideoita oli lopulta kaikkiaan yhteensä vain 15 kappaletta. Esitetyt käyttötapausideat kvantifioitiin kolmella eri viitekehyksellä, jotka eivät välttämättä ole parhaita mahdollisia VML-järjestelmän käyttötapauksien tarkasteluun. Pitää myös huomata, että kvantifioinnin teki yksi ihminen (diplomityön kirjoittaja), jonka subjektiiviset käsitykset ohjasivat käyttötapauksien luokittelua viitekehysiin.

Huomattavaa on myös, että tällä hetkellä suunnittelussa ei kuvailla tietoa abstrakteilla käsitteillä suunnitteluohjelmissa. Tällaisen tiedon kuvailu saattaa olla vieras ajatus. Kuitenkin teollisuuskumppanit korostivat työpajoissa tarvetta tunnistaa ja toteuttaa kokonaisia käyttötapauksia. He olivat myös yhtä mieltä siitä, että prosessin hallintaan vaaditaan kokonaisvaltaisempia malleja - kuten teknologiarajat ylittäviä toimintoketjukuvaavuuksia - suunnittelualueiden lokeroimisen sijaan.

VML:llä voidaan parantaa suunnitteluprosessia luomalla mekatronisen laitteen teknologiarajat ylittäviä myös abstraktia ja luonnostason suunnittelutietoa hyödynnäviä näkymiä. Tällöin voidaan auttaa teollisuutta saavuttamaan tahtotila hallita

suunnittelutietämystä näiltäkin osa-alueilta. Lopulta suunnittelun kulttuuri, työvälineet ja prosessi omaksuvat tämän kokonaisvaltaisuuden.

Vastaukset liittyivät kiinteästi olemassa oleviin kaavioihin ja suunnitteluohjelmiin. Eräs selitys voisi olla se, että nykyiset teollisen suunnittelun tavoitteet liittyvät kiinteästi lyhyen tähtäimen ideoihin ja parannuksiin. Tämä kertoo ehkä jotain suunnitteluprosessistakin: keskitytään ajallisesti paljon suunnittelun välittömien ongelmien ratkomiseen.

Kyselyssä toivottiin vastaajien esittävän kokonaisia käyttötapauksia VML:lle. Kyselyn vastauksista ei tämän vuoksi saada tietoa esimerkiksi siitä, miten VML kannattaa toteuttaa.

Esimerkiksi yhdyskuntasuunnitteluun verrattuna mekatronisesta suunnittelusta puuttuu yhteisen toimivan kokonaisuuden suunnittelunäkymä ja sen malli, jonka ympärillä yhteissuunnittelun tulisi tapahtua. Siksi tutkimusprototyypissä keskitytään ratkaisemaan yhteissuunnittelun vaatiman yhteisen työpöydän tarve, jossa tieto näytetään kaikille yhdessä samalla tavalla. Lisäksi pyritään löytämään tapoja suunnitella ja tarkastella suunnittelutyön tuloksia yhdessä vaatimusten sekä toimintojen toteutumisen näkökulmasta.

Esitetään päivitetty ominaisuusluettelo suunnittelun tukena toimivalle virtuaalikonelaboratoriolle työpajojen, kyselyn tulosten, M1-teknologioiden [44] ja yleisten virtuaalilaboratorioiden (VL) vaatimusten [16] perusteella. Taulukossa 5.1 on esitelty nämä vaatimukset selityksineen. VML:n tarvitsemat ominaisuudet on vielä ryhmitelty yleisiksi hyvien VL-ohjelmistojen ominaisuuksiksi mukaillen virtuaalilaboratorioiden tunnettuja ominaisuuksia, erityisesti VML:n ominaisuuksiksi ja mekatronisen suunnittelun tukemiseen tarvittaviksi ominaisuuksiksi (taulukossa S. tuki).

Taulukko 5.1: Yleiset vaatimukset VML-järjestelmille suunnittelun tukena.

Tyyppi	Ominaisuus ja sen toteutuksen esimerkkikuvaus.
VL	Interaktiivisuus. Vuorovaikutteinen käyttöliittymä, jonka kautta voi tutustua suunnitteluaineistoon haluamallaan tavalla.
VL	Tiedon visualisoinnit. Informatiiviset visualisoinnit. Kaksi- ja kolmiulotteiset interaktiiviset visualisoinnit yhdistettynä simulaation dynamiikkaan.
VL	Jatkokehityskelpoisuus. VML:n elinkaari pitää suunnitella siten, että järjestelmää voidaan aina jatkokehittää ja ylläpitää.
VL	Uusien teknologioiden ja standardien hyödyntäminen. VML-järjestelmä kannattaa suunnitella sellaiseksi, että sitä voidaan pitää ajan tasalla uusien teknologioiden ja standardien kanssa.
VL	Modulaarinen toteutus. Järjestelmä tulee toteuttaa modulaariseksi, jotta sen toteutuksen osia voidaan vaihtaa tarvittaessa ja kehittää itsenäisesti.
VL	Alustariippumattomuus. Toimittava useissa eri alustoissa.
VL	Edullinen toteuttaa. Tuotanto- ja ylläpitokustannukset minimoitava.
VML	Hakutoiminnallisuus. Dokumenttien, komponenttien, toimintojen ja muiden resurssien haku semanttisten hakukriteerien avulla (esimerkiksi etsi jokainen komponentti, joka osallistuu tiettyyn toimintoketjuun laitteessa).
VML	Dynaaminen reaaliaikasisimulaatio. Järjestelmän toiminnan simulointi ja simuloinnin liittäminen koneen malliin.
VML	Aidontuntuisen ohjauskontrollin antaminen laitteeseen. Laitteen aidon ohjaustuntuman tuottaminen.
VML	Simulaation hallinta. Käyttäjä haluaa mahdollisesti muuttaa simulaation toimintaa esimerkiksi esittämällä hidastettuna tietty laitteen toiminto.
VML	Yhteistoiminnalliset näkymät. Koneesta halutaan näyttää eri teknologioita yhdistäviä näkymiä, kuten esimerkiksi toimintoketjujen seurantaa. Abstraktit käsitteet helpottavat monimutkaisen monitekniikan laitteen ymmärtämistä.
S. tuki	Informaation ja aineiston yhdistely muokattavien skeemamallien avulla. VML yhdistää useista eri osa-alueista koostuvan mekatronisen laitteen kokonaisuudeksi. Skeemaa on pystyttävä muokkaamaan.
S. tuki	Datalähtöisyys. Järjestelmä määrittelee logiikan, joka tuottaa datan mukaisesti VML:n toiminnan ja näkymät. Näin VML on koko ajan ajantasaisesti suunnitteluinformaation mukainen ilman muutoksia ohjelmakoodiin.
S. tuki	Tietosisällön kommentointi ja takaisinkytkentä datalähteisiin. Suunnittelutieton kommentointi ja kytkeytyminen takaisin datalähteisiin.
S. tuki	Näkymien jakelu verkossa ja hajautettavuus. Järjestelmän on tuettava hajautusta, paikallista käyttöä ja käyttöliittymän jakelua verkossa.
S. tuki	Asynkronisuus. Suunnittelussa työtä tehdään myös asynkronisesti.
S. tuki	Testaus virtuaaliympäristössä. Mahdollisuus mitata laitteesta muuttumattomia (massa, tilavuus, pituus) ja muuttuvia suureita (paine, paikka, voima) on tärkeä ominaisuus toiminnan seurannassa.

Tutkimusprototyypin on valittu toteutettavaksi taulukon 5.1 ominaisuuksia muokailleen

- suunnittelutiedot yhdistävä käyttöliittymä,
- yhteinen työpöytä suunnittelijoille,
- vaatimusten tuominen suunnittelun pohjaksi,
- toimintojen toteuttamiseen tähtäävän suunnittelun tuki,
- ja suunnittelun kommentointitoiminnot.

Nämä ovat juuri yhteissuunnittelun kannalta tärkeitä ominaisuuksia, koska ne yhdistävät ja lähentävät eri suunnittelun osa-alueita.

Tällöin voidaan avustaa yhteissuunnittelua hallitsemaan paremmin oikeiden suunnittelukysymysten muodostaminen, yksittäisten suunnitteluratkaisujen mielekäs yhdistyminen kokonaisuuden saavuttamiseksi, suunnittelutiedon näkyväksi tuonti, jatkuvasti käytävä keskustelu suunnittelun seuraavista askelista ja yhteisen toimivan kokonaisuuden suunnittelun ylläpito.

6. TOTEUTETTU TUTKIMUSPROTOTYYPPI

Suurin VML:n tarjoama hyöty saadaan kaikkien suunnittelijoiden yhteistyön auttamisessa yhteissuunnittelussa. Modernitkin CAD-suunnitteluvälineet saattavat välillä jopa haitata suunnittelutiedon välisten yhteyksien hahmottamista [58]. Tällöin jää piiloon esimerkiksi sellaiset tärkeät asiat, kuten laitteen vaatimusten ja toimintojen toteutuminen sekä niiden dokumentointi [51].

Tutkimusprototyyppi on rakennettu webin teknologioin ja se hyödyntää erityisesti semanttisen mallin avulla linkittyvää suunnittelutietoa. Webin teknologiat tarjoavat parhaat käytänteet tällaisten monimuotoisten tietojen linkittämiseen keskenään. Tällöin VML ei myöskään ota kantaa käytettäviin suunnitteluohjelmiin vaan tietomalli perustuu avoimiin standardeihin.

Kuitenkin pitää muistaa, että nykyiset suunnitteluohjelmat eivät integroidu hyvin keskenään [58] [51]. Työvälineiden toimintatavat periytyvät piirto-ohjelmista ja tiedon tallennuksen lopputulos vaikuttaa olevan usein PDF tai muu säiliö, jonka sisältämä tieto ei ole koneluettavaa tai linkittyntä muuhun suunnittelutietoon. Ikävä kyllä myöskään PDM-järjestelmät eivät ratkaise tällä hetkellä tiedon yhdistelyn ongelmia, koska niitä käytetään kuten dokumenttitietokantoja. Mahdollisesti eräs vaihtoehto olisi ohjelmistojen työprosessin hyvin täsmällinen suunnittelu.

Tutkimusprototyypin avulla on tarkoitus tutkia, miten informaatio saadaan käsiteltyä siten, että suunnittelutiedon generoitavissa olevan VML:n yhdistävä käyttöliittymä luo yhteisen näkymän suunnittelutietoon. Samalla tuodaan vaatimukset ja toimintoketjut osaksi semanttista suunnitteluprosessia ja näkyviin VML-ohjelmistoon.

Tutkimusprototyypissä VML:n tarvitsemaa semanttista mallia varten rakennettiin sovittimia ja tiedonkäsittelyn putkilinjoja. Näin saatiin esiteltyä ne hyödyt, jotka ovat saavutettavissa, kun suunnittelutieto on linkittynyt keskenään. Näiden ratkaisujen avulla voidaan tarkastella, millaista suunnittelutietoa pitää tuottaa, jotta VML saadaan toimimaan yhteissuunnittelun tukityövälineenä.

- **2. Miten VML-järjestelmä kannattaa toteuttaa suunnittelun tueksi?**
 - Miten saadaan VML toimimaan datalähtöisesti?
 - Millaista tietoa VML-järjestelmä vaatii?
 - Millaisia aineistovaatimuksia näkymät asettavat?

- Miten tietomallia hallitaan?
- Miten simulaatio liitetään VML-näkymiin?
- Miten VML kytkeytyy takaisin suunnitteluprosessiin?

Tutkimusmateriaalina on poralaitteen puomin valmis kaaviomateriaali, jonka avulla on rakennettu toimiva puomi. Kuitenkin tunnetusti nämä nykyiset suunnitteluinformaation kuvaukset eivät sisällä riittävää informaatiota VML:n toteuttamiseksi. Onkin mielekästä tarkastella sitä, mitä koneluettavaa informaatiota suunnittelumateriaalin pitäisi sisältää, jotta yhteissuunnittelua ja toimivan puomin rakentamista voitaisiin tukea tietokoneavusteisesti.

Puomin suunnittelua pyritään parantamaan ottamalla käyttöön tiedon yhdistämistä ja tuottamista helpottavia menetelmiä sekä työvälineitä. Näiden avulla saadaan koneluettavaksi sellaista tietoa, jota voidaan hyödyntää VML-näkymien generoinnissa.

Mitattavia asioita ovat:

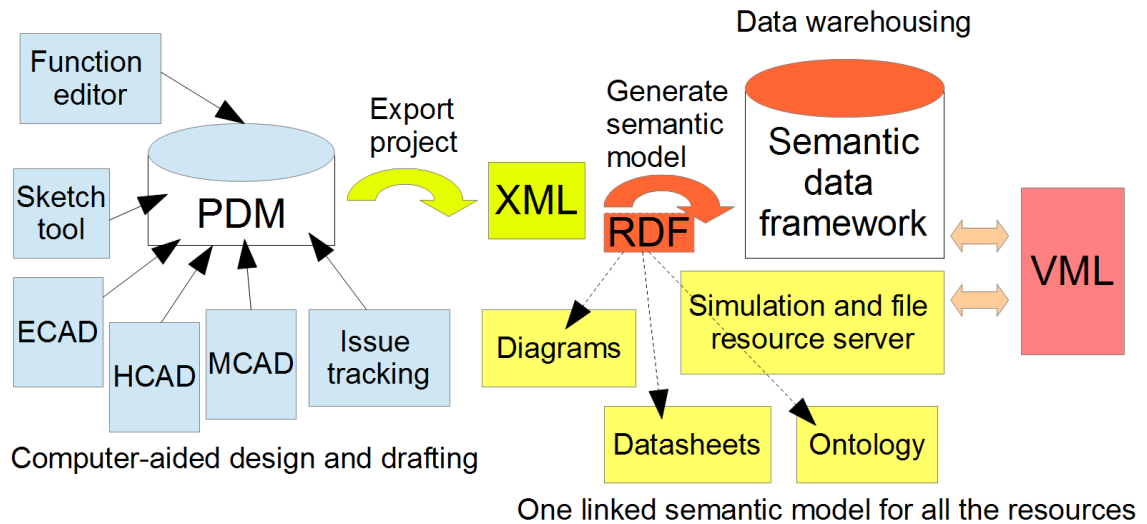
- koneluettavan suunnitteluinformaation määrä;
- kuinka hyvin tuotettu tieto tukee kokonaisen koneen vaatimusten tarkastelua;
- ja informaation läpäisy ja yhtenäisyys putkilinjassa.

Generoitavien VML:n näkymien avulla voidaan havainnoida tarpeita ja puutteita suunnittelutiedossa; suunnitteluohjelmilla tuotetun suunnittelutiedon perusteella pitäisi olla mahdollista esittää virtuaalinäkymä toimivasta puomista ja selittää perustellusti sen toimivia suunnitteluratkaisuja.

6.1 Toteutustapa

Prototyyppiin toteutettu suunnitteluinformaation käsittely tapahtuu kuvassa 6.1 yksinkertaistetusti esitetyllä tavalla. Suunnitteluohjelmissa oli tuotettu suunnittelutietoa, jonka Semogen-hankkeen jäsenet tallensivat PDM-järjestelmään. Kun projektin putkilinja suoritetaan, niin PDM:stä luetaan projekti XML-muodossa. Sitten XML-muotoinen tieto käännetään semanttiseksi malliksi eli tässä RDF/XML:ksi. Tämä semanttinen tietomalli linkittää yhteen kaiken suunnittelutiedon ja erilliset suunnitteludokumentit. Semanttiseen tietomalliin päätellään lisää yhteyksiä OWL-ontologian avulla. Päätelty RDF/XML-tieto lisätään semanttisen datan sovelluskehys Callimachukseen, minne on mahdollista luoda datalähtöisiä näkymäpohjia koneen suunnittelutiedosta. Nämä Callimachuksen näkymät yhdessä jatkokehitetyn Semoplayerin kanssa luovat VML:n koostesovelluksena.

Callimachuksen vastuulla on semanttisen datan hallinta, kyselyrajapinnat ja semanttisen mallin mukainen navigointi. Semogen:n ensimmäisen osan toteutuksesta



Kuva 6.1: Suunnitteluinformaation käsittely suunnitteluohjelmista semanttiseen malliin ja sitä kautta VML-järjestelmään. Suunnitteluohjelmilla tuotettu tieto tallentuu PDM-järjestelmään, josta se siirretään putkilinjaan XML-muodossa. Tämä muoto muunnetaan semanttisen tietomallin RDF/XML-formaattiin, joka yhdistelee suunnitteluaineiston. Jatkokokehitetty Semoplayer ja semanttisen datan sovelluskehys muodostavat yhdessä VML:n koostenäkymät suunnittelutietoon.

jatkokehitetty Semoplayer taas hoitaa suunnitteludokumenttien jakelun sekä simulaation tilan jakelun WebSocket-rajapinnan kautta. Semoplayer:n vastuulla on myös joystick-ohjauskontrollit laitteeseen ja dynaamisten muutosten esittäminen suunnittelukaavioissa sekä 3D-malleissa.

6.2 Toteutetut ominaisuudet

Diplomityössä on esitelty ominaisuusluettelo taulukossa 5.1 suunnittelun tukena toimivalle virtuaalikonelaboratoriolle työpajojen, kyselyn tulosten, M1-teknologioiden ja yleisten VL-järjestelmien vaatimusten perusteella.

Näistä ominaisuuksista toteuttamatta tutkimusprototyyppi-VML:ssä jäi vain simulaation hallinta siltä osin, että sitä voitaisiin käyttää muuten kuin reaaliaikaisimulaation suorittamiseen. Tosin ominaisuudet ovat toteutettu vain tutkimus- ja testitarkoituksessa puomimateriaalille eivätkä ne sinällään ole valmiita esimerkiksi tuotantokäyttöön.

Seuraavaksi käydään läpi, miten nämä ominaisuudet ovat demonstroituna toteutetussa VML:ssä. Kukin ominaisuus esitellään lyhyesti ja esitellään sen mahdollinen toteutus esimerkki prototyypistä.

Interaktiivisuus: Vuorovaikutteinen käyttöliittymä, jonka kautta voi tutustua suunnitteluaineistoon haluamallaan tavalla. VML:n prototyypin käyttöliittymä on toteutettu kokonaan selaimen HTML5:n ja JavaScriptin avulla.

Tiedon visualisoinnit: Informatiiviset visualisoinnit. Kaksi- ja kolmiulotteiset

interaktiiviset visualisoinnit yhdistettynä simulaation dynamiikkaan. Kaaviot esitetään prototyypissä SVG-kuvina ja 3D-grafiikka X3D-malleina, joihin on liitetty interaktiivista toiminnallisuutta JavaScriptin avulla.

Jatkokehityskelpoisuus: VML:n elinkaari pitää suunnitella siten, että järjestelmää voidaan aina jatkokehittää ja ylläpitää. Modulaarisuus, datalähtöisyys ja valmiiden standardien käyttö helpottavat jatkokehitystä ja ylläpitoa. Esimerkiksi tutkimusprototyyppiin voi vaihtaa puomin tilalle jonkin toisen mekatronisen laitteen RDF-tietomallin, SVG-kaaviot ja X3D-mallit ilman, että käyttöliittymän ohjelmakoodia joudutaan muuttamaan.

Uusien teknologioiden ja standardien hyödyntäminen: VML-järjestelmä kannattaa suunnitella sellaiseksi, että sitä voidaan pitää ajan tasalla uusien teknologioiden ja standardien kanssa. Prototyypissä käytetään laajasti uusia teknologioita ja standardeja, kuten selaimen ominaisuuksista WebSocket-tietoliikenerajapintaa; 3D-grafiikalle WebGL-grafiikkarajapintaa; HTML5-standardin ominaisuuksia (kuten canvas-piirto-pinta) ja JavaScript-ohjelmointikielen Worker-säikeistystä rinnakkaisuuden toteuttamiseksi.

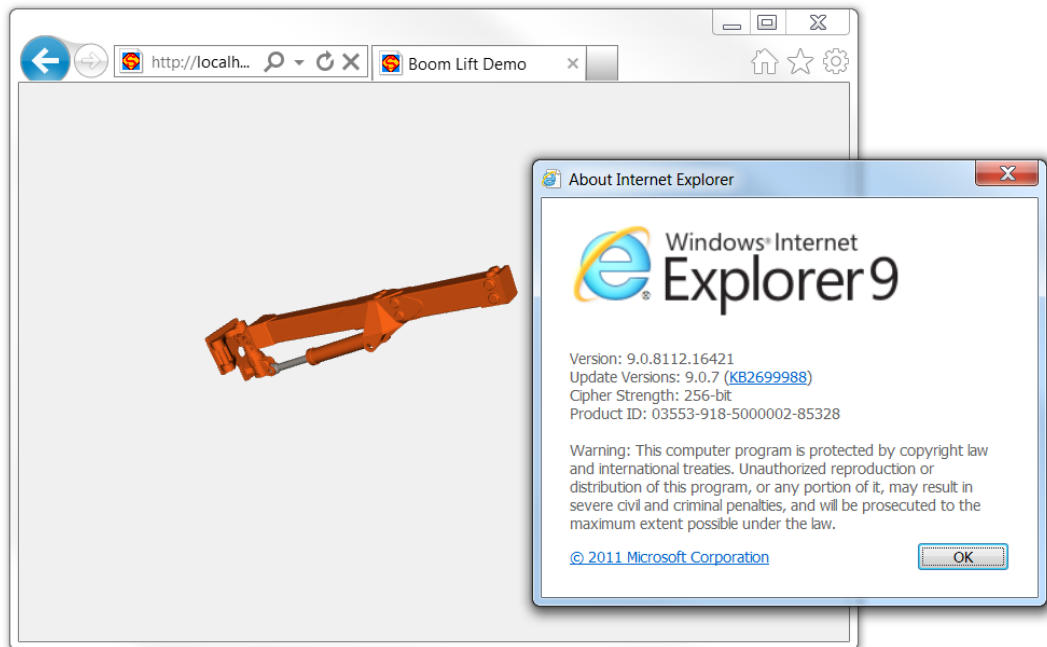
Säikeistystä asiakaspäässä (selain) tarvitaan siihen, että simulaation kuuntelu WebSocket-yhteydellä ei tuki käyttöliittymän sulavaa toimintaa. Rinnakkaisuuden toteutus löytyy liitteistä.

Tallennusmuotojen ja informaation hallinnan kohdalla uusien standardien hyödyntäminen näkyy 3D-mallin formaattina käytetyssä avoimessa X3D-formaatissa, RDF-tietomallissa, SPARQL-kyselykielen uusimman 1.1-version käytöllä ja kuvien SVG-grafiikkana.

Modulaarinen toteutus: Järjestelmä tulee toteuttaa modulaariseksi, jotta sen toteutuksen osia voidaan vaihtaa tarvittaessa ja kehittää itsenäisesti. Tutkimusprototyyppi on hyvin modulaarinen, koska se koostuu useista täysin itsenäisesti toimivista osista, kuten simulaatio, sovittimet sekä putkilinja, semanttisen tiedon hallintajärjestelmä ja käyttöliittymä useine erilaisine näytinkomponentteineen. Prototyyppiä toteutettaessa komponentteja vaihdeltiin tarpeen mukaan: putkilinjaa ja sovittimia muokattiin toistuvasti ja alkuperäisen Semogen-hankkeen ensimmäisen vaiheen Semoplayerin sisäinen semanttinen datan hallintakomponentti vaihdettiin Callimachus-järjestelmäksi. Simulaation voisi toteuttaa käytetyn Matlab Simulink-ohjelmiston sijaan muillakin tavoilla, kunhan toteuttaa simulaation tilan lähettämiseksi vaaditun tietoliikenerajapinnan ja sen kuvailun generoinnin.

Lisäksi prototyypissä saman toiminnallisuuden toteuttaa useampi täysin erillinen toteutusteknologia. Esimerkiksi, jos käytetään selainta, joka ei toteuta teknologiastandardeja, niin prototyyppisovellus osaa tehdä useita yhteensopivuuden ja toimivuuden varmistavia toimenpiteitä (fall-back). Jos selain ei tue säikeistystä, niin käyttöliittymä ei toimi rinnakkaisesti. Jos WebSocket-tuki ei ole olemassa, niin siir-

retään simulaatitieto HTTP:llä. Jos selain ei tue WebGL-grafiikkaa, niin 3D-malli näytetään Flash:n avulla. Näiden varmistusten ansiosta VML toimii jopa Internet Explorer-selaimessa, mikä ei tarjonnut Semogen-hankkeen toteutusajankohtana tukea rinnakkaisuuteen, 3D-grafiikkaan tai WebSocket-yhteyksille (kuvassa 6.2).



Kuva 6.2: Prototyypissä on useita yhteensopivuuden varmistavia teknologioita, jotka pitävät huolen siitä, että VML:n käyttöliittymä toimii myös selaimissa, joissa uusimmat standardit eivät ole toteutettuna. Kuvassa Internet Explorer -selain, jossa simulaation tahdissa liikkuva 3D-malli näytetään Flash-tuen ja HTTP-tiedonsiirron avulla uusimpien WebGL- ja WebSocket-standardien sijaan.

Alustariippumattomuus: Toimittava useissa eri alustoissa. Selaimen käyttö näyttiminä mahdollistaa sen, että VML:n käyttöliittymää voidaan käyttää missä tahansa järjestelmässä, jossa toimii moderni selainohjelma. Sen sijaan prototyypin simulaatiokomponentti (.exe) tarvitsee Windows-käyttöjärjestelmän ajoympäristökseen tai yhteensopivuuskerroksen Linux-käyttöjärjestelmissä (esim. Wine). Muu osa prototyypin palvelintoteutuksesta toimii täysin ainakin Linux- ja Windows-käyttöjärjestelmissä.

Edullinen toteuttaa: Tuotanto- ja ylläpitokustannukset minimoitava. Datalähtöisyys pienentää tuotantokustannuksia, koska prototyypin sisältöä voidaan muuttaa vaihtamalla tietomalli. Sen lisäksi, lukuun ottamatta simulaatiota, kaikki VML:n osat käyttävät ilmaisia vapaasti saatavilla olevia ohjelmistokirjastoja.

Hakutoiminnallisuus: Dokumenttien, komponenttien, toimintojen ja muiden resurssien haku semanttisten hakukriteerien avulla (esimerkiksi etsi jokainen komponentti, joka osallistuu tiettyyn toimintoketjuun laitteessa). RDF-tietomalli mal-

lintaa laitteen siten, että SPARQL-kyselyrajapinnan kautta on yksinkertaista toteuttaa mikä tahansa kysely suunnittelutiedolle. Tietomallia on rikastettu semanttisen prosessin mukaisesti erilaisin menetelmin ja työvälinein siten, että se sisältää tietoja, joita ei nykyisessä koneensuunnittelussa kerätä täsmällisesti talteen. Tästä esimerkkinä laitteen toimintojen suunnittelu.

Dynaaminen reaaliaikasisimulaatio: Järjestelmän toiminnan simulointi ja simuloinnin liittäminen koneen malliin. Reaaliaikasisimulaation tutkimusryhmä toteutti Matlab Simulink -ohjelmistolla. Simulaatiopalvelin käynnistetään prototyypin palvelimen mukana ja se siirtää simulaation tilatiedon UDP-paketteina prototyypin HTTP- ja WebSocket-yhteydet toteuttavalle komponentille ja niin edelleen asiakkaalle selaimen 6.5.

Aidontuntuisen ohjauskontrollin antaminen laitteeseen: Tutkimusryhmä ohjelmoi joystick-kontrollit simulaatioon, joten sitä voidaan kontrolloida reaaliaikaisesti. Puomiin saatiin aito ohjaustuntuma.

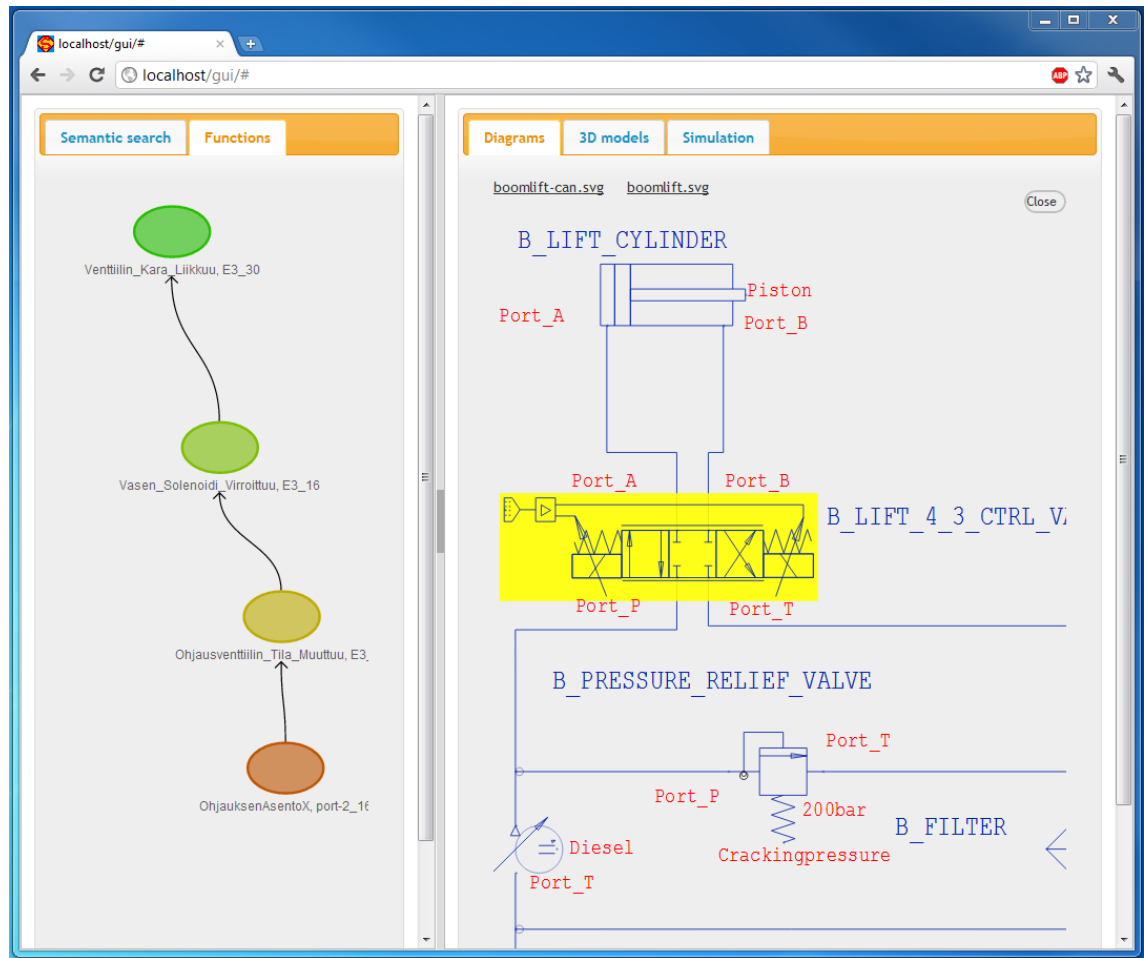
Simulaation hallinta: Käyttäjä haluaa mahdollisesti muuttaa simulaation toimintaa esimerkiksi esittämällä hidastettuna tietty laitteen toiminto. Tällaista ominaisuutta ei prototyypissä ole toteutettuna.

Yhteistoiminnalliset näkymät: Koneesta halutaan näyttää eri teknologioita yhdistäviä näkymiä, kuten esimerkiksi toimintoketjujen seuranta. Abstraktit käsitteet helpottavat monimutkaisen moniteknisen laitteen ymmärtämistä. Toimintoketjut on mallinnettu osaksi semanttista RDF-tietomallia ja niiden tarkastelu on mahdollista VML:n käyttöliittymän kautta (kuvassa 6.3).

Informaation ja aineiston yhdistely muokattavien skeemamallien avulla: VML yhdistää useista eri osa-alueista koostuvan mekatronisen laitteen kokonaisuudeksi. Skeemaa on pystyttävä muokkaamaan. Semanttiselle mallille luotiin useita skeema- ja ontologiakuvauksia OWL- ja RDFS-teknologiolla. Nämä yhdistävät suunnittelutietoa kokonaisuuksiksi.

Datalähtöisyys: Järjestelmä määrittelee logiikan, joka tuottaa datan mukaisesti VML:n toiminnan ja näkymät. Näin VML on koko ajan suunnitteluinformaation mukainen ilman muutoksia ohjelmakoodiin. Semogen-hankkeen toisen puoliskon aikana pyrittiin muuttamaan prototyyppi mahdollisimman datalähtöiseksi sovellukseksi; kovakoodattua dataa poistettiin näyttimestä ja korvattiin ohjelmalogiikalla, joka generoi tietomallista näyttimen.

Tietosisällön kommentointi ja takaisinkytkentä datalähteisiin: Eräitä prototyypin datalähteitä ovat semanttiset datalehdet, jotka luodaan ja hallitaan prototyypin semanttisessa sovelluskehyksessä Callimachusessa. Datalehtiä voidaan paitsi luoda, niin myös muokata ja kommentoida sovelluskehyksessä. Vastaavasti olisi mahdollista määritellä mille tahansa semanttisen mallin osalle muokkaus- ja kommentointinäkymät.



Kuva 6.3: Esimerkki prototyypin toimintoketjun seuraamisesta. Toimintoketjun palanen ja hydraulikaavio vierekkäin. Toimintoketjun suunnittelu ja ylläpito on osa semanttista suunnittelua.

New Check Valve

Model
CHVFL16WF

Manufacturer
TPP

Description
Check valve 2-pos, size 16, for water-based fluids.

Price
220

External Resources

Image URL
<http://smartsimu.rd.tut.fi>



View Edit Discussion Describe History

exampleproject » semogen » datasheet » sheets » checkvalve » CHVFL16WF

CHVFL16WF 239 euros TPP

Check valve 2-pos, size 16, for water-based fluids.

Technical information

Weight	Maximum Operating Pressure	Type	Mounting
3	250	ISO-flange	
Flow at 10bar Pressure Loss	Number of Ports	Number of Positions	Mounting
220	2	2	ISO-flange

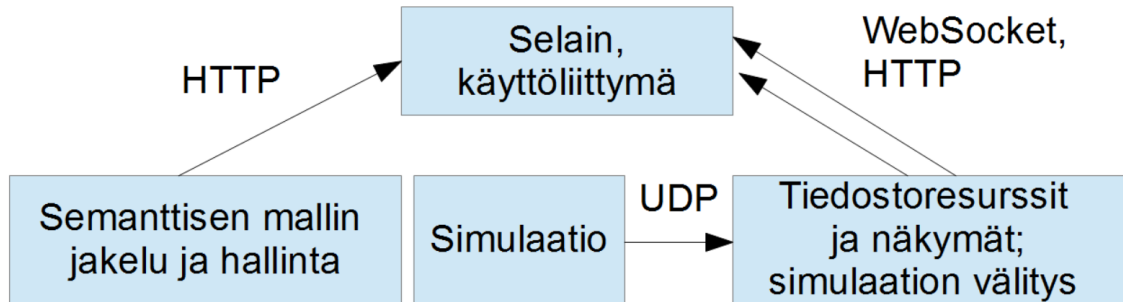
Fluids: • water
• HFA
• HFC

Resources • [homepage](#) • [data](#)
• [symbol](#) • [mfile](#)

Kuva 6.4: Semanttisen datalehden luonti ja tarkastelu. Tiettyyn datalehteen liittyy semanttinen malli, katselunäkymä, editointinäkymä, keskustelu ja muokkaushistoria. Datalehden käyttöoikeuksia ja näkyvyyttä voidaan myös hallita.

Liitteenä kuvat semanttisen datalehden koneluettavasta semanttisesta puolesta ja ihmisluettavasta puolesta.

Näkymien jakelu verkossa ja hajautettavuus: Järjestelmän on tuettava hajautusta, paikallista käyttöä ja käyttöliittymän jakelua verkossa. Järjestelmä on hajautettu asiakas-palvelin-mallisesti. Asiakasohjelmalla toimii selainohjelmisto ja palvelin koostuu useista erillisistä palveluista, jotka voidaan vielä periaatteessa hajauttaa kukin omalle tietokoneelleen (simulaatio, Callimachus-sovelluskehys sekä HTTP- ja WebSocket-palvelun toteuttava komponentti, kuvassa 6.5).



Kuva 6.5: Asiakas-palvelin-mallinen prototyypin toteutus. Selain kytkeytyy palveluihin ja tarjoaa käyttöliittymän VML:llään. Jokainen komponentti voidaan suorittaa omalla tietokoneella.

Asynkronisuus: Suunnittelussa työtä tehdään myös asynkronisesti. Koko semanttinen prosessi, putkilinja ja VML:n käyttötapa tukevat asynkronista työskentelyä.

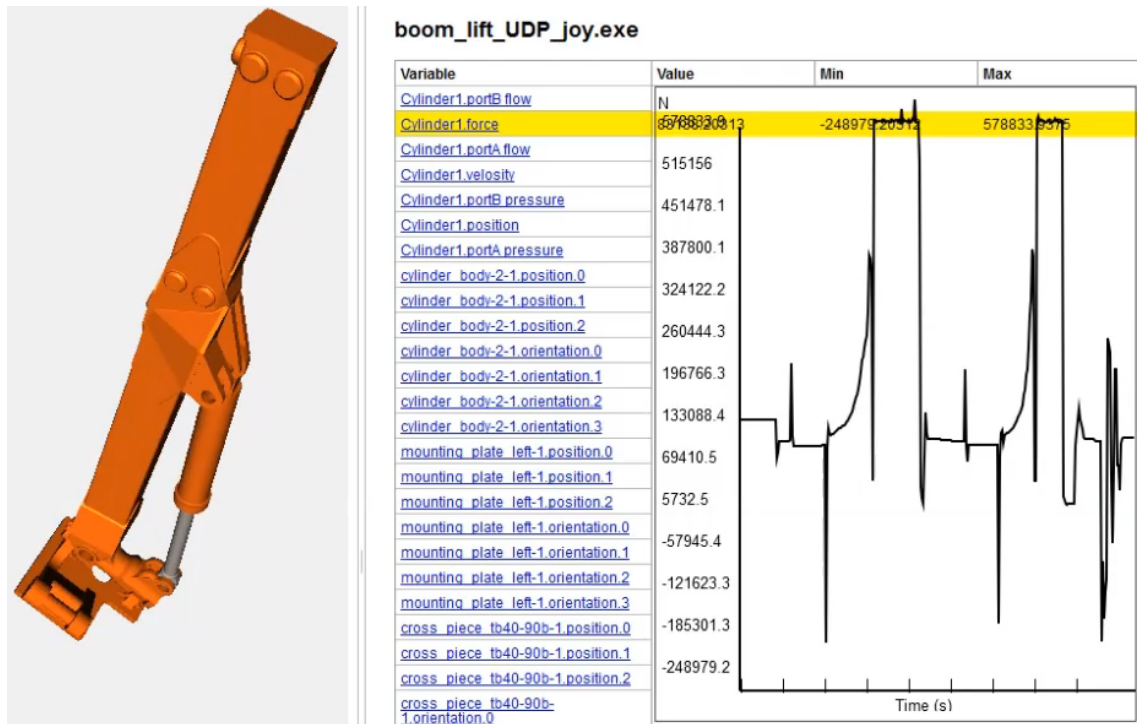
Testaus virtuaaliympäristössä: Mahdollisuus mitata laitteesta muuttumattomia (massa, tilavuus, pituus) ja muuttuvia suureita (paine, paikka, voima) on tärkeä ominaisuus toiminnan seurannassa. Simulaatiosta tulevien muuttuvien suureiden käyttäytymistä voi seurata VML:n käyttöliittymässä. Kuvassa 6.6 on esitelty, kuinka prototyypissä voi tehdä mittauksia eri simulaation suureista ja samalla tarkkailla 3D-näyttimestä puomin käyttäytymistä.

Mittarin piirturi käyttää HTML5-standardin canvas-piirto pintaa esittämään tietyn simulaation arvon käyttäytymistä.

Toteutettujen ominaisuuksien perusteella prototyypin toteutustekniikat vaikuttavat lupaavilta ja toimivilta. Webin tekniikat soveltuvat sellaisenaan VML:n toteutustavaksi.

6.2.1 Vaatimusten ja toimintomallin hallinta

Vaatimusten toteutumisen seuranta tuodaan osaksi VML-järjestelmää lukemalla korjaus- ja muutospyyntöjen hallintajärjestelmästä tiedot semanttiseen malliin. Kun muutkin semanttisen mallin tiedot, myös vaatimusten toteuttamiseen tähtäävä työtehtävien hallinta esimerkiksi tikettien hallinnalla on silloin näkyvillä VML:ssä.



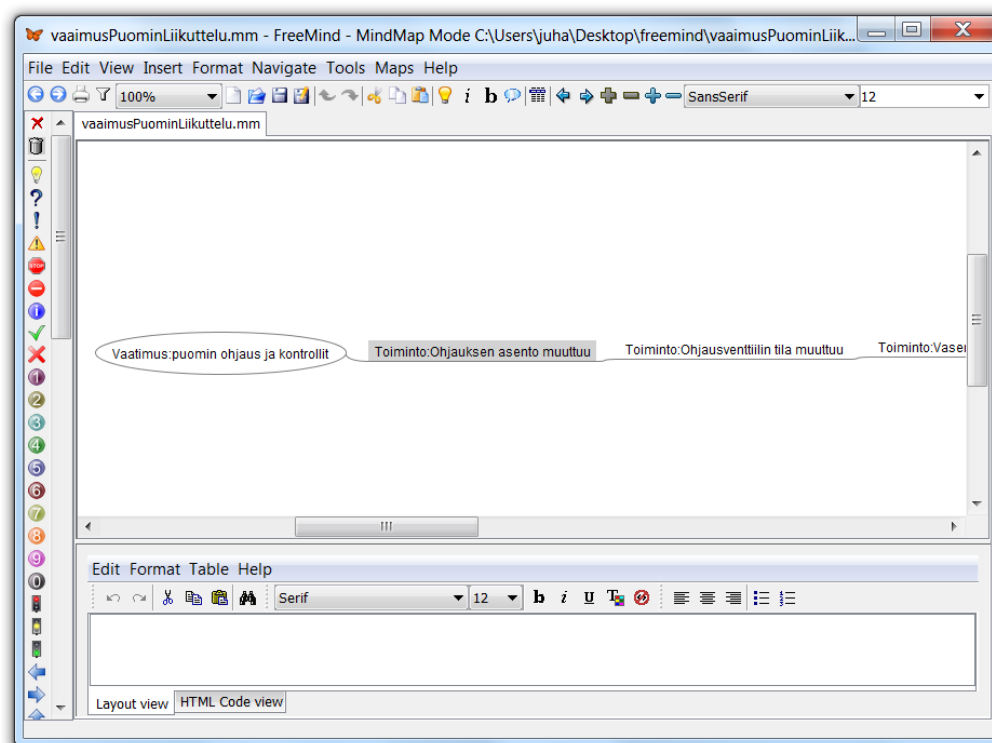
Kuva 6.6: VML:n prototyyppissä on toteutettuna simulaation tilan mukainen 3D-mallin käyttäytyminen ja simulaation tuottamien muuttuvien suureiden arvojen käyttäytymisen seuranta piirturissa.

Luonnollisesti vaatimusten määrittelystä seuraa koneen toimintojen luonnostelu, joka sekä halutaan koneluettavaksi mukaan semanttiseen malliin ja edelleen VML-järjestelmään näkyville.

Järjestelmän toteuttamat toiminnot kuvataan toimintomalliksi, joka kertoo järjestelmän komponenttien välisen vuorovaikutuksen. Toimintomallien hallintaan on testattu ratkaisuksi työkalua, jolla toiminnot voidaan suunnitella ja lukea osaksi semanttista mallia. Tällöin nekin ovat mukana VML:ssä ja muodostavat mahdollisuuden navigoida suunnitteluaineistoa toimintojen näkökulmasta.

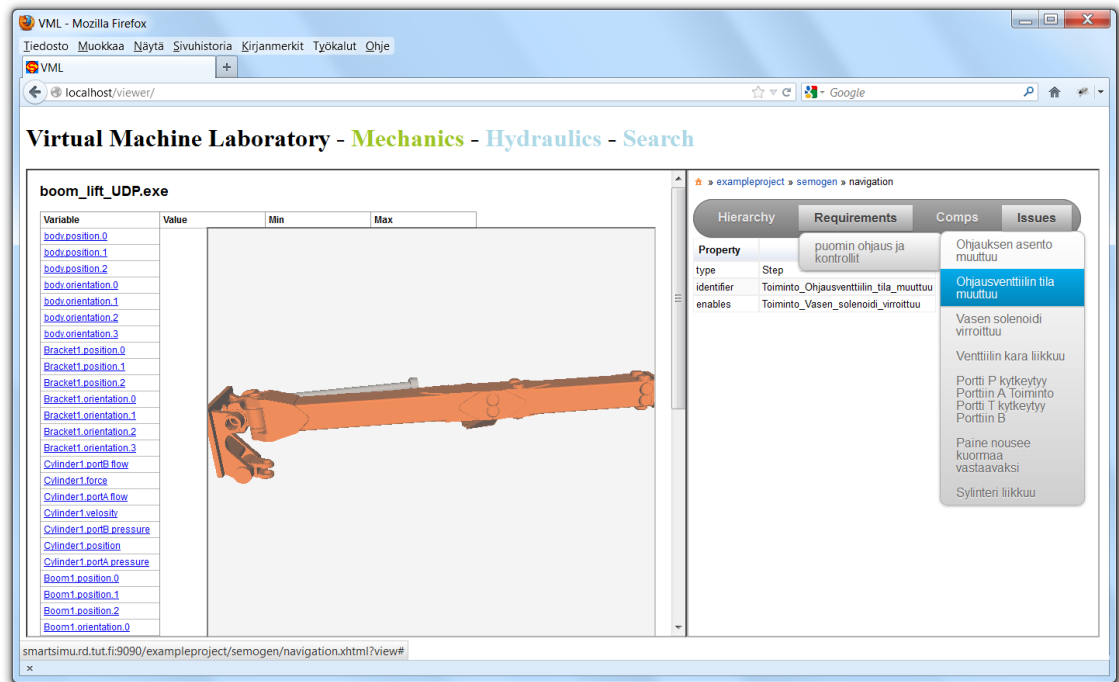
Ilmaiseen Freemind-käsittekarttaeditoriin¹ lisättiin sovitinohjelma, joka generoi käsittekarttana luodun toimintomallin RDF:ksi. Näin editoria voi käyttää suunnittelun ylätasoa mallina toimivien toimintoketjujen suunnittelussa. Kuvassa 6.7 on esimerkki toimintojen suunnittelusta editorilla.

¹Freemind-käsittekarttaeditori, http://freemind.sourceforge.net/wiki/index.php/Main_Page (24.11.2012)



Kuva 6.7: Ilmaisella Freemind-käsittekarttaeditorilla voi suunnitella toimintoja. Suunnittelussa ylätasoinen mallina toimiva toimintomalli on myös semanttisessa mallissa eri suunniteltua tietoa yhdistävä ylätasoinen malli. Jotta käsittekartan tieto saadaan osaksi semanttista mallia, kirjoitettiin Freemind-editoria varten sopiva sovitinohjelma.

Tallennettu käsittekartta muunnetaan sovitinohjelman avulla RDF:ksi, kun putkilinjassa määritellyt generoinnit suoritetaan. RDF-muodossa tallennettu semanttinen tietomalli lisätään Callimachus-järjestelmään, jossa on määritelty osa VML:n näytelmien generoinnista vastaavasta toteutuksesta. Kuvassa 6.8 näkyy VML:n valikko, jossa on suunnittelussa määritellyt toiminnot laitteelle.



Kuva 6.8: Koneen suunnitteluinformaation selailu toimintojen avulla. Valikko on generoitu VML:llään semanttisesta mallista.

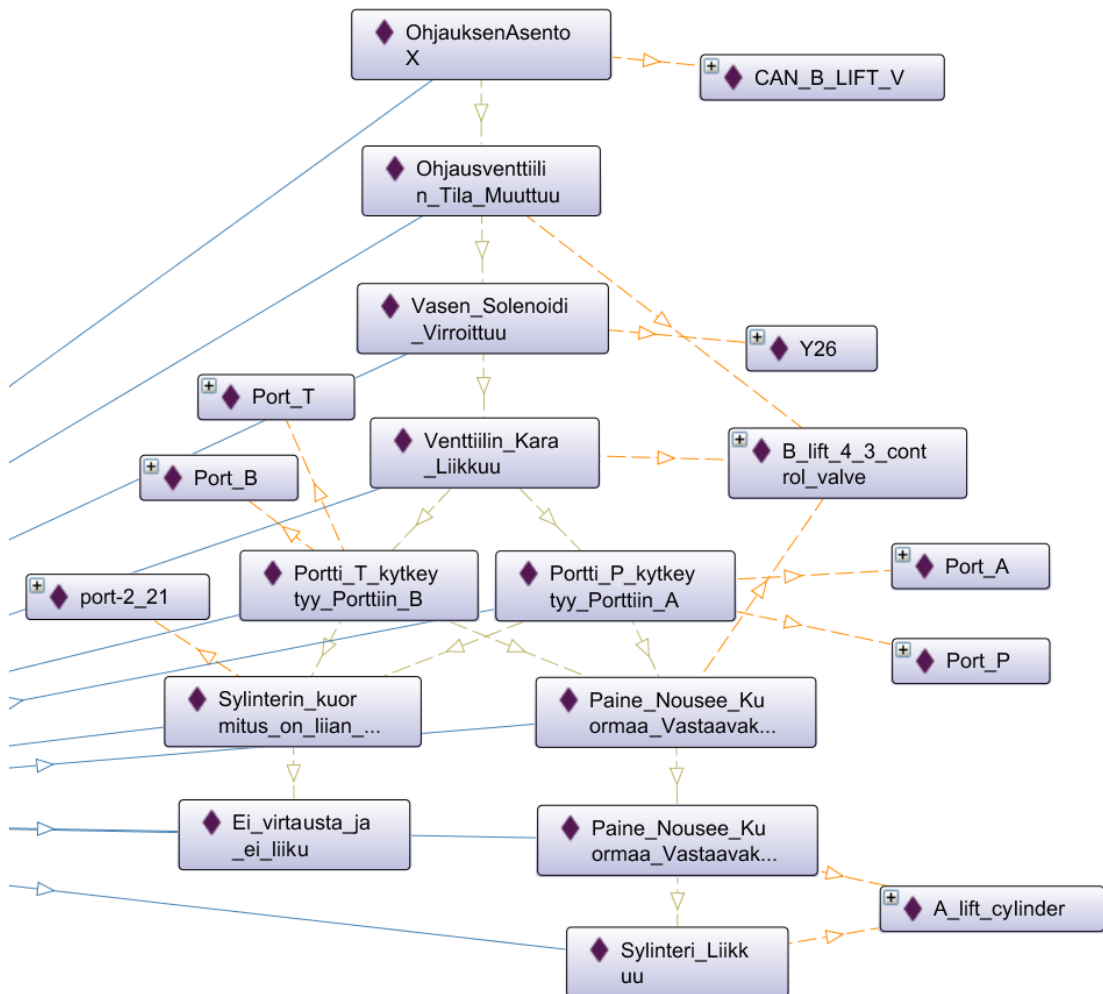
Toimintomalliin linkittyy koko projektin suunnittelutieto. Samalla se on vaihtoehtoinen tapa mallintaa konetta järjestelmäkuvauksen rinnalla. VML:ssä on mahdollista tarkastella suunnittelutietoa navigoimalla sitä toimintojen näkökulmasta.

Järjestelmän toiminnot kuvaa käsitekarttaeditorista generoitu semanttinen RDF-malli. Toimintojen malli mallintaa toiminnot osakohtaisesti ja mahdollistaa monihaaraisen tapahtumaketjun. OWL-mallinen ontologia kuvaa, millainen on semanttinen toimintomalli.

Toiminnot mahdollistavat toisia toimintoja ja muodostavat toimintojen verkon. Toimintojen verkossa kuljetut polut muodostavat lopulta toimintoketjun. Toimintoon liittyvä komponentti vaikuttaa tilallaan siihen, mitkä ovat toiminnon laukaisemina toimintoja.

Ajatuksena on, että heti suunnittelun alkuvaiheessa tehdään luonnostason toimintomalli, jota koko suunnittelun ajan tarkennetaan lähemmäs täsmällistä toteutusta tietyillä teknisillä ratkaisuilla. Näin suunnittelulla on koko ajan käytössä ylätasoinen tietomalli, johon voidaan liittää kaikki laitteeseen tulevat järjestelmät ja selittää niiden tehtävä laitteessa. Tämän avulla suunnittelua voidaan tarkastella VML:ssä yhteissuunnittelun mukaisesti kaikkia suunnittelun osa-alueita koskevan toimintojen ratkaisemisongelman näkökulmasta.

Kuvassa 6.9 on toimintomalli avattuna Protégé-nimisellä suositulla ja ilmaisella ontologioiden hallintatyökalulla. Kuten kuvasta näkyy, niin toimintomalli linkittää eri järjestelmiä yhteen.



Kuva 6.9: Yksittäistä toimintoketjua kuvaava semanttinen malli visualisoituna Protégé-nimisessä ohjelmassa. Kuvasta nähdään, millaiset askeleet toimintoketjussa on. Askeliin on linkittynyt järjestelmät, jotka toteuttavat tietyn toiminnon laitteissa.

Validi RDF/XML- ja OWL/XML-kuvaus esimerkkitoimintoketjusta löytyy liitteistä, jossa samassa tiedostossa on OWL-ontologiamalli toimintoketjulle sekä esimerkkinä yhden toimintoketjun kuvaus RDF-muodossa.

6.3 Generoinnin vaatimukset

Virtuaalisen konelaboratorion on oltava koko ajan generoitavissa automaattisesti suunnittelutiedosta, jotta sitä voidaan hyödyntää suunniteltavan kokonaisuuden ymmärtämisessä. Näin VML kykenee näyttämään ajan tasaisen koneen virtuaaliprototyyppin kaikille suunnitteluun osallistuvilla. [36]

Generointi asettaa vaatimuksia tuotetulle suunnittelutiedolle: suunnittelutiedon on oltava yhdistettävissä eri suunnitteluohjelmista yhtenäiseksi malliksi ja semanttisen kuvailun on oltava sillä tarkkuudella, että tietokone osaa hyödyntää suunnittelutietoa. [36]

Suunnitteluohjelmistojen pitää pystyä siirtämään suunnittelutieto yhteiseen tietomalliin, josta voidaan generoida näkymät koneen prototyyppiin. Semogen-hankkeessa toteutettiin puomimateriaalille ja hankkeessa käytetyille suunnittelua tukeville työvälineille putkilinja, jonka avulla tieto virtasi suunnitteluohjelmista semanttiseen kokonaisuon. [42]

6.3.1 Putkilinjakuvaus

Semogen-hankkeessa määriteltiin Apache Ant -ohjelmistolla niin sanottu putkilinja (englanniksi pipeline), jossa määritellään prosessi, joka kutsuu projektissa toteutettuja eri suunnitteluohjelmistojen tietoja varten luotuja sovittimia ja suunnittelutiedon käsittelyohjelmistoja. Lopputuloksena putkilinja tuottaa semanttisen tietomallin. [42]

Tällä automaattisella prosessilla tieto siis virtaa suunnittelijoilta suunnitteluohjelmistojen välityksellä aina tietomalliin, josta virtuaalinäkymät koneeseen voidaan generoida.

Olisi toivottavaa, että suunnitteluohjelmistot tukisivat suoraan tiedon tuontia semanttiseen malliin sekä helpottaisivat tiedon yhdistelyä ja tuottamista suunnittelijoiden kesken.

6.3.2 Tietomallin vaatimukset

Semanttisen tietomallin vaatimus on, että sillä voidaan kuvata kaikki suunnittelutieto ja tiedon väliset yhteydet koneluettavaan muotoon siten, että tietokone voi käsitellä automaattisesti koko koneen suunnitteludokumentaation. Koska suunnittelussa tuotetaan kaikki se tieto, jota tarvitaan laitteen rakentamisessa, niin mallintamalla tämä tieto koneluettavaan muotoon tietokone osaa rakentaa virtuaalisesti laitteen jo keskeneräisestä suunnittelutiedosta.

Jotta virtuaalinen prototyyppi voidaan esittää jo suunnittelun alussa, tietomallissa on oltava heti alkuvaiheessa luonnos ylätasoin suunnittelutiedosta, jolla voidaan näyttää jonkinlainen käsitys siitä, minkälaisen rungon ympärille koneen suunnittelutieto alkaa tarkentumaan. Konkreettisesti tämä voi olla luonnos laitteen 3D-mallista ja siihen liittyvät toiminnot, jotka vaatimusten perusteella osataan hahmotella laitteelle.

Yhteissuunnittelun kannalta on tärkeää, että tietomalli on aina ajan tasalla ja suunnittelijat näkevät toistensa työn vaikutuksen suunniteltavassa kokonaisuudessa.

sa. Ajan tasaisen tiedon perustella, jonka tietokone osaa vielä yhdistellä ja näyttää, voidaan tehdä yhteisiä suunnitteluratkaisuja ja iteroida virtuaalisen prototyypin avulla.

Vaatimukset tietomallille tulevat niistä käyttötapauksista, joita suunnitteluryhmä tarvitsee työnsä helpottamiseksi. Toisaalta sellaista lisätyötä ja tiedon rikastamista ei kannata tehdä, joka ei maksa itseään takaisin prosessin aikana.

Minimivaatimuksena voisi olla, että suunnittelutieto tuotetaan sillä tarkkuudella, että yksittäistä komponenttia koskeva tieto on liitetty yhteen, löydettävissä ja esitettävissä sekä liitetty koko laitetta mallintaviin runkomalleihin, kuten tietoon siitä, mitä toimintoja komponentti on mukana toteuttamassa ja missä järjestelmässä se on osana. Komponentin semanttinen datalehti sisältää taas yksityiskohtaiset tiedot siitä, millainen komponentti on kyseessä.

6.3.3 Vaatimukset suunnitteluohjelmistoille

Jotta semanttinen tietomalli voidaan tuottaa automatisoidusti ja sitä kautta generoida VML-järjestelmään näkymä suunnittelusta, suunnitteluohjelmistojen pitää tukea semanttisesti kuvaillun suunnittelutiedon tuottamista. [42]

Toisaalta suunnitteluohjelmistojen pitää tehdä mahdollisimman helpoksi tietojen yhdistäminen esimerkiksi semanttisessa mallissa jo oleviin laitteen toimintoihin.

Tarvitaan myös uudenlaisia suunnitteluohjelmistoja, kuten prototyyppituotteen toimintojen suunnitteluun tarkoitettu editori. Tällaisia suunnittelun apuvälineitä on mielekästä lisätä siinä määrin, että niiden tuottama hyöty on suurempaa kuin niiden tuottamien tietojen ylläpitämisen vaiva. Erityisesti pitää muistaa, että suunnittelijan pitäisi pystyä tekemään suunnittelutyötään suoraviivaisesti ilman useilla työkaluilla tehtävää ylläpitotyötä.

Suunnitteluohjelmistojen käytössä on myös toivomisen varaa. Suunnitteluohjelmistot eivät ole pelkkiä piirtotyökaluja vaan niitä tulisi käyttää ennen kaikkea suunnittelutiedon luontiin, yhdistelyyn ja hallintaan.

Tavoitteena on lopulta juurikin suunnittelijoiden työmäärän vähentäminen. Semanttinen suunnitteluprosessi pitää integroida osaksi suunnitteluprosessia ja sen työvälineitä siten, että esimerkiksi tehty luonnostelutyö ja luonnosten tarkenteluun käytetty työmäärä vähentääkin lopulta kokonaistyömäärää.

7. ARVIOINTI JA YHTEENVETO

7.1 Vertailu muihin virtuaalilaboratorioihin

VML-tutkimusprototyypin on toteutettu yhteissuunnittelua varten erilaisia ominaisuuksia. Toteutettua prototyyppiä vertaillaan yhdeksään muuhun virtuaalilaboratoriojärjestelmään tarkastellen yleisten VL-järjestelmien vaatimusten toteutumista. Nämä vaatimukset ovat 1) uusien teknologioiden ja standardien hyödyntäminen; 2) reaaliaikasisimulaatio; 3) näkymien jakelu tietoverkossa; 4) alustariippumattomuus; 5) edullinen toteutus; 6) jatkokehityskelpoisuus ja sisällön muokkaamisen tuki; 7) asynkronisuus; 8) tarkoituksenmukaisuus; 9) tuki kieliversioinnille; 10) kehittäminen käyttäjäseurannan avulla; 11) modulaarinen toteutus; 12) interaktiivisuus; 13) tietosisällön kommentointi; 14) hakutoiminnallisuus; 15) simuloinnin kontrollointi ja 16) käyttäjän laatimien testien esittäminen [16].

Juuri tämä vaatimuslista on valittu, koska on olemassa valmista vertailututkimusta VL-järjestelmistä käyttäen juuri näitä vaatimuksia [16]. Näin ollen tutkimusprototyypin laajempi vertailu helpottuu käyttäen jo muuallakin käytettyä vaatimuskriteeristöä sekä sen avulla tehtyjä ominaisuuksien VL-kohtaisia toteumatarkasteluja.

Vaikka vaatimuslista on suunniteltu alunperin opetuskäyttöön tarkoitettuja virtuaalilaboratorioita varten, niin se on hyvin samankaltainen M1-teknologioiden mukaisten ominaisuuksien kanssa ja soveltuu myös suunnittelun tukityövälineeksi suunnitellun virtuaalisen konelaboratorion (yhdeksi) vaatimuslistaksi. Näistä vaatimuksista nykyinen tutkimusprototyyppi toteuttaa vaatimukset 1-9 ja 11-15 eli vain kaksi kuudestatoista jää toteutumatta.

Arviointimenetelmänä on yleiskäyttöinen MCDM eli multiple-criteria decision-making, jota voidaan soveltaa myös ohjelmistojen vertailuun. MCDM-menetelmä on tarkoitettu käytettäväksi, silloin kun vertaillaan erilaisia järjestelmiä keskenään. Tällöin etsitään järjestelmien vertailukriteerit ja annetaan vertailukriteereille painokertoimet. Kriteereille annetaan jokaisen järjestelmän kohdalla lukuarvot, jotka kerrotaan painokertoimilla ja summataan yhteen. Näin saadaan valittujen kriteerien ja painokertoimien mukainen numeerinen lukuarvo jokaisesta järjestelmästä. Menetelmä johtaa läpinäkyvään päätöksen tekoon ja vertailuongelman täsmällistämiseen, koska päätöksenteolla on tietyt näkyvät kriteerit ja painokertoimet. [43]

Verrokkeina tutkimusprototyypin arviointiin on yhdeksän olemassa olevaa VL-järjestelmää, joiden toteuttamat vaatimukset tunnetaan [16]. Tämän lisäksi vertai-

luun on lisätty näillä vaatimuksilla paras mahdollinen teoreettinen järjestelmä, joka toteuttaa kaikki vaatimukset sekä huonoin mahdollinen järjestelmä, joka ei toteuta mitään vaatimuksista. Tällainen menetelmä tunnetaan parhaan ja huonoimman tapauksen analyysinä [13]. Kaikkiaan keskenään vertailtavia järjestelmiä on siis kaksitoista kappaletta.

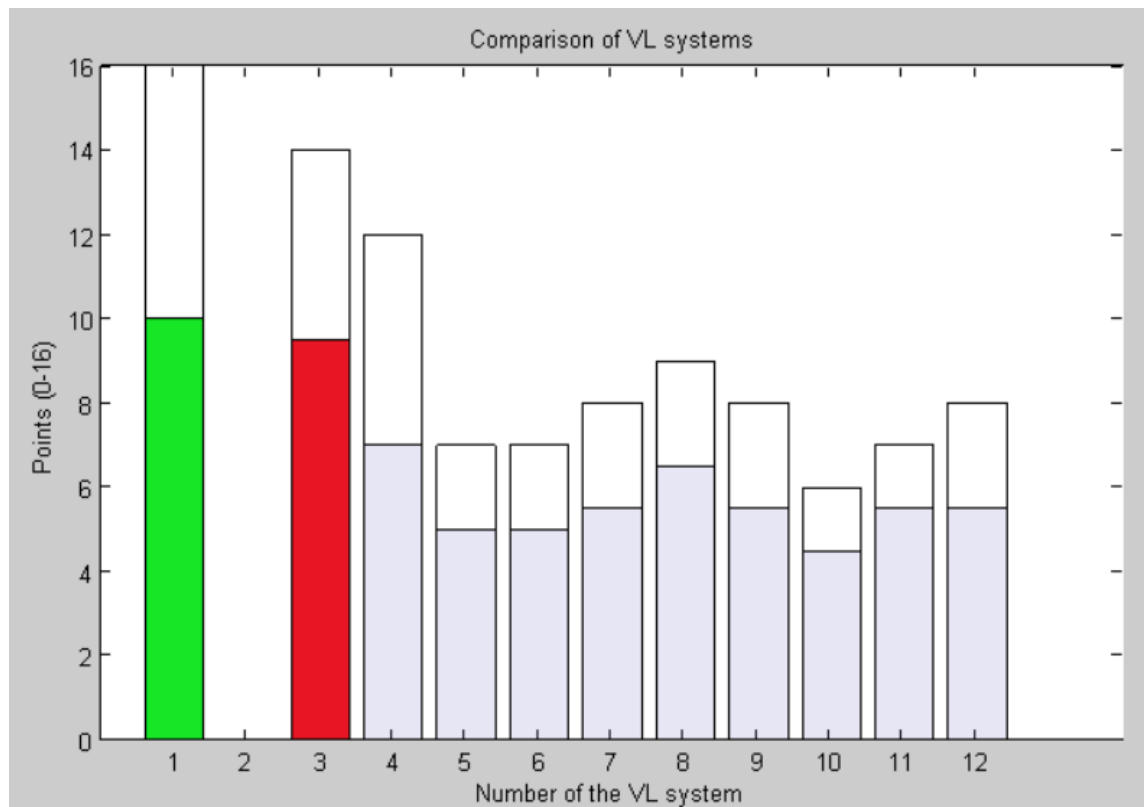
1. Kaikki vaatimukset täyttävä verrokkijärjestelmä.
2. Ei mitään täyttävä verrokkijärjestelmä.
3. Tässä projektissa tuotettu tutkimusprototyyppi.
4. PhET, Physics Education Technology:
<http://phet.colorado.edu/new/index.php>
5. Johns Hopkins University VL:
<http://www.jhu.edu/virtlab/virtlab.html>
6. VCLab, Virtual Control Lab:
<http://www.esr.ruhr-unibochum.de/VCLab/index.html>
7. Physics Applets:
<http://jersey.uoregon.edu/>
8. LabVir:
<http://personales.upv.es/jogomez/labvir/>
9. Laboratorio Virtual de Física:
<http://www.enciga.org/taylor/lv.htm>
10. VLEE, Virtual Laboratory for Earthquake Engineering:
<http://cee.uiuc.edu/sst1/java/>
11. OWL, The Online Widener Laboratories:
<http://quantum.soe.widener.edu:280/>
12. Virtlab, A Virtual Laboratory:
<http://www.virtlab.com/main.aspx>

Vaatimuslistan vaatimukset on painotettu sen mukaisesti, kuinka välttämättömiä vaatimukset ovat suunnittelun tukemisen kannalta. Nämä painokertoimet on valittu kyselyn, M1-teknologioiden asettamien vaatimusten ja tutkimusprototyyppiä kehitettäessä jalostuneiden havaintojen perustella. Vaatimusten painokertoimet voivat saada selkeyden vuoksi kolmea eri diskreettiä arvoa: painokerroin on 0, 0.5 tai

1, jossa painokerroin 0 tarkoittaa ei-välttämätöntä, 0.5 tärkeää ja 1 välttämätöntä vaatimusta.

Tässä diplomityön tekijä on tehnyt luokittelun suunnittelua varten. Kun painokertoimet on annettu järjestyksessä jokaiselle vaatimukselle, on saatu painokerroinvektori, jolla on painokertoimet (0.5, 0.5, 1, 0.5, 0.5, 1, 1, 0, 0.5, 0, 1, 1, 0.5, 1, 0.5, 0.5).

Tällä on kerrottu binäärinen ominaisuusmatriisi, jossa on riveittäin vertailujärjestelmien ominaisuuksien toteutuminen asteikolla toteutuu (1) ja ei toteudu (0). Tulokseksi on saatu lukuarvot, jotka kertovat kuinka hyvin tämän analyysin mukaan järjestelmät toteuttavat vaatimukset. Järjestelmien vertailu on esitetty värillisinä palkkeina kuvassa 7.1. Kuvassa näkyy myös valkoisina taustapalkkeina järjestelmien saamat tulokset silloin, jos jokaiselle vaatimukselle on annettu painoarvo 1 eli välttämätön. Tällainen vertailu on johdettu herkkyysanalyysistä [13].



Kuva 7.1: VL-järjestelmien vertailu vaatimusten suhteen. Järjestelmä yksi on kaikki kriteerit toteuttava täydellinen verrokkijärjestelmä. Järjestelmä kaksi ei toteuta mitään. Tutkimusprototyyppi on 3. järjestelmä. Opetuskäyttöön suunnitellut verrokkijärjestelmät ovat vaaleat palkit (4-12).

Koska tutkimusprototyyppiä kehitetyssä VML:ssä suurin osa (14/16) näistä yleisistä VL-järjestelmien vaatimuksista on toteutettuna, niin tutkimusprototyyppi pärjää vertailussa erittäin hyvin. Yksi syy tähän on, että prototyypissä haluttiin testata hyvin laajasti VML:n ominaisuuksien toteuttamista. Toisaalta pitää muistaa, että

tutkimusprototyyppi on ominaisuuksien toteutuksen laajuudeltaan suppeampi kuin verrokkijärjestelmät. Tutkimusprototyyppi ei myöskään sellaiseen sovellu tuotantokäyttöön vaan käsittelee suppeaa puomimateriaalia tutkimusmielessä.

Tulos kertoo, että valituilla webin tekniikoilla ja datalähtöisellä toteutustavalla voidaan ketterästi toteuttaa virtuaalikonelaboratoriossa tarvittavia ominaisuuksia. Tällä toteutustavalla voi toteuttaa kokonaisen VML:n, joka generoidaan semanttisesta mallista datalähtöisesti. On mielekästä kokeilla rakentaa tuotteistettu VML käyttäen tiedon mallintamiseen ja hallintaan semanttisen webin tekniikoita (RDF, OWL, SPARQL) ja selainta käyttöliittymän tekemiseen (HTML5, WebGL, JavaScript).

7.2 Yhteenveto

Suunnittelijoilta puuttuu tällä hetkellä tietokoneavusteinen yhteinen suunnittelu-pöytä, jonka ääressä yhteissuunnittelu tapahtuisi tuetusti. Tällaisen ympäristön toteuttamiselle on havaittu selkeä tarve. Eräs tällainen yhteissuunnittelua auttava työväline on virtuaalinen konelaboratorio (VML), jossa on toteutettuna suunnittelutiedon yhdistäminen eri suunnittelun osa-alueiden välillä. Tavoitteena on realistinen käsitys siitä toimilaitteesta, joka tuotetun suunnittelutiedon perusteella on synty-mässä.

Jotta VML voitaisiin tuottaa automatisoidusti ja datalähtöisesti suunnittelutiedosta ilman erillistä VML-ohjelmistoprojektia suunnitteluprojektin rinnalla, tarvitaan suunnittelutiedon yhdistävä tietomalli ja automatisoitu tietomallin muodostaminen. Ikävä kyllä suunnitteluohjelmistot ja PDM-järjestelmät eivät yhdistele suunnittelutietoa yhtenäiseksi malliksi. Tällaiselle suunnittelutietoa yhdistävälle mallille on suuri tarve paitsi VML:n generoinnin kannalta, niin myös yleisesti helpottamaan koneensuunnittelua tietokoneavusteisesti.

Koneensuunnitteluohjelmistojen ongelmat liittyvät suljettuihin formaattijärjestelmiin ja rajapintoihin, jotka estävät tietojen siirron ja yhdistelyn eri suunnittelu-alojen välillä. Ratkaisu tähän on yhteiset rajapinnat ja siirtotavat kaikille suunnitteluohjelmistoille siten, että suunnittelutieto yhdistyisi saumattomasti yhteen. [36] Nykyisistä teknologioista semanttinen web tukee erittäin hyvin informaation jakelua ja yhdistelyä, joten sen hyödyntäminen on mahdollista koneensuunnittelutietojen yhdistelyssä. Yhteisten rajapintojen ja formaattien kanssa on mahdollista päästä eroon tilanteesta, jossa olisi pakko käyttää saman ohjelmistotalon tekemiä tuotteita ja niiden vaihtaminen olisi haastavaa (vendor lock-in) [39].

Toinen koneensuunnitteluvälineiden ongelma on niiden käytössä, jolla saataan tähdätä vain ihmisluettavuuteen (paperidokumentaatioon). CAD-ohjelmistoja käytetään usein piirto-ohjelmina, vaikka niiden oleellisempi hyöty saavutetaan silloin, kun niitä käytetään suunnittelutiedon tuottamiseen ja yhdistelyyn tietokonea-

vusteisesti. Samoin PDM-järjestelmien tärkein tehtävä on hallita suunnittelutietoa sinällään eikä vain kerätä talteen dokumentteja ja dokumenttien metatietoa. Tässäkin semanttisen webin toimivimpia teknologioita kannattaa soveltaa.

Semogen-hankkeessa kehitetty semanttinen suunnitteluprosessi tuottaa koneluettavaa suunnittelutietoa. Tämä suunnittelutieto on tuotettu sitä varten, että lopulta helpotettaisiin ihmisten suorittamia suunnittelutyönvaiheita automatisoinnilla ja suunnittelutiedon uudelleen käytöllä. [36] Vaikka semanttinen suunnittelu monimutkaistaa suunnittelutiedon käsittelyä tietokoneilla, niin tavoitteena on saada suunnittelu sujumaan suunnittelijoiden näkökulmasta yksinkertaisemmin.

Parantamalla suunnitteluvälineiden ja sitä kautta tuotetun suunnittelutiedon yhteensopivuutta saavutetaan parempi yhteissuunnittelun tuki, kun suunnittelutieto liikkuu ihmiseltä toiselle. Tavoitteena on siirtää tietokoneen tehtäväksi monia automatisoitavia tehtäviä suunnittelusta ja näin ihmisten tekemän käsityön määrä vähenee. Tällöin myös koneluettavalle suunnittelutiedolle voidaan muodostaa yhteinen virtuaalinen näkymä, jonka ympärillä suunnittelijat voivat tarkastella sitä kokonaisuutta, jonka heidän tuottama suunnittelutietonsa yhdessä tuottaa.

Yhteissuunnittelussa suunnittelijat voivat keskustella koneesta kokonaisena toimivana laitteena. Ideaalitulanteessa jokainen suunnittelija tekee oman vastualueensa CAD-ohjelmistoilla muutoksia suunnittelutietoon ja tallentaa muutokset kaikkien nähtävälle koneen virtuaaliseen prototyyppiin. Tämän virtuaalinäkymän ympärillä yhteissuunnittelu kokoontuu tarkastelemaan yhteistä suunnittelutietoa. Tällöin voidaan tarkastella erilaisten teknisten ratkaisujen yhteensopivuutta ja toimivuutta sekä ennen kaikkea sitä, miten suunnittelulle asetetut vaatimukset ovat toteutumassa. Koko suunnittelutyö voidaan nähdä suunnittelun osa-alueiden sisäisten vaatimusten toteuttamisen sijaan yhteisten vaatimusten toteuttamisena.

Puomin suunnitteluaineistoa yhdistettäessä Semogen-hankkeessa havaittiin, että yhteissuunnittelun kannalta keskeisiä vaatimus- ja toimintojen suunnittelutietoja ei ole dokumentoitu näkyviin [42]. Yhteissuunnittelun näkökulmasta toiminnot muodostaisivat luontaisen yhteisen rungon eri osa-alueiden suunnittelutiedolle. Diplomi-työssä kokeiltu tapa yhdistellä suunnittelutietoja toimintojen mallilla on hyvin luontainen tapa rakentaa koneen ylätasoa malli. Tällainen toimintomalli voidaan luoda heti suunnittelun luonnosteluvaiheessa ja mallia voidaan ylläpitää koko tuotteen elinkaaren ajan. Tähän malliin linkittyy kaikki muu suunnittelutieto ja se selittää suunnittelutiedon tarkoitusta laitteessa.

Suunnittelutieto siiloutuu suuressa määrin eri suunnittelun teknisten osa-alueiden sisällä. Yhteissuunnittelua varten kannattaisi tutkia ja toteuttaa sellaista tietokoneavusteista sovellusta, jolla voidaan luoda heti suunnittelun alussa laitteen toimintomalli ja ylläpitää sitä koko suunnittelun ajan sekä testata tämän mallin käytön hyötyjä. Mekatronisen koneen suunnittelutieto olisi tällöin aina osa tämän mallin

suunnitelmaa ja kaikella tällä tiedolla olisi aina perusteltu merkitys.

VML:n jatkokehitys semanttisen webin teknologioilla tuottaisi suunnittelun rinnalle lisää generoitavissa olevia näkymiä. Asiakas-palvelin-mallinen VML, jossa käyttöliittymä toteutetaan selaimen, on hajautettava, prototyypin avulla perustellusti toteutettavissa nykyisillä tekniikoilla, ei vaadi käyttäjän koneelta kuin selainohjelman, ketterä toteuttaa, useita avoimia standardeja tukeva, modulaarinen ja jatkokehityskelpoinen. VML asettaa vastaavia vaatimuksia webin teknologioille kuin nykyaikaiset tietokonepelit: suorituskykyä, liikkuvaa 3D-grafiikkaa, reaaliaikaisia tapahtumia ja sisällön vaivattoman lisäämisen datalähtöiseen näyttimeen.

Eryteisesti simulaation avulla joystickillä ohjattavissa oleva interaktiivinen puomi asettaa suuret vaatimukset selaimen uusille standardeille. Se kuitenkin on mahdollista toteuttaa ja kertoo siitä, että koko VML-järjestelmä tai muu vastaavat vaatimukset omaava järjestelmä voidaan toteuttaa täysin nykyisten avoimien standardien mukaisesti ja se toimii suorituskyvyltään keskinäisen tietokoneen selaimessa.

Jos VML toteutetaan tuotteeksi käyttäen webin tekniikoita, niin web-ohjelmointi tuo haasteita ohjelmistoprojektille. Nämä haasteet liittyvät siihen, että webin teknologiat ja niiden käyttötavat muuttuvat hyvin nopeasti ja niillä ei ole vakiintunutta ohjelmistotuotannon perinnettä [30].

Webin tekniikoilla toteutetun VML:n kannalta suunnittelutieto on helposti hyödynnettävissä, kun se on siirretty avoimien standardien mukaiseen muotoon (prototyypissä X3D ja SVG) ja yhdistelty semanttisella mallilla (prototyypissä RDF, OWL). VML on esimerkkijärjestelmänä siitä, miten suunnitteluaineistoa voidaan yhdistellä ja yhteissuunnittelua tukea.

LÄHTEET

- [1] J. Adolfsson ja J. Karlsen. *Mechatronics 98*. Pergamon., 1998. s. 940.
- [2] M. Airila. *Mekatroniikka*. Hakapiano Oy., Helsinki, 1999. s. 69.
- [3] K. Al-Kodmany. Using visualization techniques for enhancing public participation in planning and design: process, implementation, and evaluation. *Landscape and Urban Planning*, 45:37–45, 1999.
- [4] K. Al-Kodmany. Disco: a co-design online tool for asynchronous distributed child and adult design partners. *Proceedings of the 11th International Conference on Interaction Design and Children*, ss. 11–19, 2012.
- [5] D. Allemang ja J. Hendler. *Semantic Web for the Working Ontologist, Modeling in RDF, RDFS and OWL*. Morgan Kaufmann Publishers Inc., USA, 2008. s. 330.
- [6] S. Battle, D. Wood, J. Leigh, ja L. Ruth. The callimachus project: Rdfa as a web template language. *Third International Workshop on Consuming Linked Data, COLD2012*, 2012.
- [7] R. Beaker, J. Grudin, W. Buxton, ja S. Greenberg. *Readings in Human-Computer Interaction: Toward the Year 2000*. Morgan Kaufmann Publishers Inc., San Francisco, toinen laitos, 1995. s. 950.
- [8] T. Berners-Lee. Semantic web - xml2000 by tim berners-lee. w3c. <http://www.w3.org/2000/Talks/1206-xml2k-tbl/slide10-0.html>, 2000. [WWW] [viitattu 14.07.2012].
- [9] M. Böhms, D. Leal, H. Graves, ja K. Clark. Product modelling using semantic web technologies. w3c, incubator group report 08 october 2009. <http://www.w3.org/2005/Incubator/w3pm/XGR-w3pm-20091008/>, 2009. [WWW] [viitattu 29.06.2012].
- [10] R.H. Bishop. *Mechatronics: An Introduction*. Taylor & Francis, 2006.
- [11] D Brickley ja R. V. Guha. Rdf primer. w3c, recommendation 10 february 2004. <http://www.w3.org/TR/2004/REC-rdf-primer-20040210/>, 2004. [WWW] [viitattu 30.06.2012].
- [12] T. Clark ja M Ocana. Semantic web applications in neuromedicine (swan) ontology. w3c, interest group note 20 october 2009. <http://www.w3.org/TR/hcls-swan/>, 2009. [WWW] [viitattu 29.06.2012].

- [13] European Commission. Econometrics and applied statistics, sensitivity analysis. joint research centre - sensitivity analysis. <http://ipsc.jrc.ec.europa.eu/?id=752>, 2012. [WWW] [viitattu 08.11.2012].
- [14] Web3D Consortium. Web3d - what is x3d?. <http://www.web3d.org/about/overview/>, 2012. [WWW] [viitattu 26.12.2012].
- [15] Kenneth Crow. Product data management and product information management. drm associates. <http://www.npd-solutions.com/pdm.html>, 2002. [WWW] [viitattu 21.11.2012].
- [16] J. Cuartero-Olivera ja A. Perez-Navarro. Virtual laboratories and their implementation. *EAAEIE Annual Conference*, ss. 1–6, 2009.
- [17] P2P Foundation. Co-design. <http://p2pfoundation.net/Co-Design>, 2011. [WWW] [viitattu 08.01.2013].
- [18] Google. Google documents. google docs home. <http://www.google.com/google-d-s/intl/en/tour1.html>, 2011. [WWW] [viitattu 27.10.2012].
- [19] The Khronos Group. WebGL specification editor's draft 07 march 2011. <http://www.khronos.org/registry/webgl/specs/latest/>, 2011. [WWW] [viitattu 25.12.2012].
- [20] I. Haikala ja J. Märijärvi. *Ohjelmistotuotanto*. Talentum Media Oy, 11. laitos, 2006. s. 440.
- [21] S. Harris ja A. Seaborne. Sparql 1.1 query language. w3c, working draft 05 january 2012. <http://www.w3.org/TR/sparql11-query/>, 2012. [WWW] [viitattu 20.07.2012].
- [22] M. Helminen, T. Palonen, M. Rokala, P. Ranta, T. Mäkelä, ja K. Koskinen. Virtual machine laboratory based on ml-technology. *Proceedings of the Twelfth Scandinavian International Conference on Fluid Power*, 1:321–334, 2011.
- [23] HTML5. Html5 a vocabulary and associated apis for html and xhtml. <http://dev.w3.org/html5/spec/Overview.html>, 2011. [WWW] [viitattu 23.12.2012].
- [24] 3 Round Stones Inc. Callimachus project. project's homepage. <http://callimachusproject.org/>, 2012. [WWW] [viitattu 20.07.2012].
- [25] Ecma International. Standard ecma-262 ecmascript language specification. <http://www.ecma-international.org/publications/standards/Ecma-262.htm>, 2009. [WWW] [viitattu 24.12.2012].

- [26] I. Kant. The critique of pure reason. <http://www2.hn.psu.edu/faculty/jmanis/kant/Critique-Pure-Reason.pdf>, 1781. [WWW] [viitattu 28.10.2012] s. 565.
- [27] A. Korttemaa. Metviro-oppimisympäristön vianhaun käytettävyyden kehittämien. *Diplomityö, Tampereen teknillinen yliopisto, Tampere*, 2008. . s. 82.
- [28] T. Lehtonen, T. Juuti, H. Oja, S. Suistoranta, A. Pulkkinen, ja A. Riitahuhta. A framework for developing viable design methodologies for industry. *International Conference on engineering design, ICED11*, 1:405–416, 2011.
- [29] F. Manola ja E. Miller. Rdf vocabulary description language 1.0: Rdf schema. w3c, recommendation 10 february 2004. <http://www.w3.org/TR/rdf-schema/>, 2004. [WWW] [viitattu 12.07.2012].
- [30] T. Mikkonen ja A. Taivalsaari. WebGL seminar 2011. <http://lively.cs.tut.fi/seminars/WebGL2011/WebGL-seminar-2010-1209.pdf>, 2011. [WWW] [viitattu 25.12.2012].
- [31] J. Nurmi. Kallionporauksen 3d-visualisointi selainteknologioidilla. *Kandidaatintyö, Tampereen teknillinen yliopisto, Tampere*, 2011. . s. 24.
- [32] O. Nykänen. Rakenteiset dokumentit -kurssi, kevät 2009. hypermedian opetuksen kotisivu. <http://matwww.ee.tut.fi/hmopetus/rd/2008/arc/rd08-196-212.pdf>, 2009. [WWW] [viitattu 20.07.2012].
- [33] O. Nykänen. Semanttiset tekniikat ja sovellukset -kurssi, syksy 2011. ebookbrowser. <http://ebookbrowse.com/nykanen-sta-2011-s01-s07-v2011-10-07-pdf-d311647312>, 2011. [WWW] [viitattu 12.01.2013].
- [34] O. Nykänen, K. Koskinen, P. Ranta, S. Salonen, M. Rokala, M. Helminen, J. Nurmi, H. Sairiala, V. Alarotu, J. Aaltonen, ja S. Pohjolainen. Semantic top-down modeling of mechatronics systems for sustainable product data and lifecycle management. *In Proceedings of Mechatronics 2012*, ss. 792–799, 2012.
- [35] O. Nykänen, J. Salonen, ja J. Huhtamäki. Wille visualisation toolkit for developers with a concise oks visualisation application catalogue for end-users. *Deliverable D10.18. OPAALS project (IST-034824).*, 2010.
- [36] O. Nykänen, J. Salonen, M. Markkula, P. Ranta, M. Rokala, M. Helminen, V. Alarotu, J. Nurmi, T. Palonen, K. Koskinen, ja S. Pohjolainen. What do information reuse and automated processing require in engineering design? *Journal of Industrial Engineering and Management*, 4:669–698, 2011.

- [37] University of Colorado. Phet: Free online physics, chemistry, biology, earth science and math simulations. <http://phet.colorado.edu/>, 2008. [WWW] [viitattu 31.12.2012].
- [38] C. Ogbuji. Sparql 1.1 graph store http protocol. w3c, working draft 1 may 2012. <http://www.w3.org/TR/sparql11-http-rdf-update/>, 2012. [WWW] [viitattu 20.07.2012].
- [39] OSLC. Open services for lifecycle collaboration. <http://open-services.net/>, 2012. [WWW] [viitattu 21.11.2012].
- [40] E. Prud'hommeaux ja A. Seaborne. Sparql query language for rdf. w3c, recommendation 15 january 2008. <http://www.w3.org/TR/rdf-sparql-query/>, 2008. [WWW] [viitattu 20.07.2012].
- [41] A. Raaphorst ja R Johnson. Dita open toolkit user guide. oasis. <http://dita-ot.sourceforge.net/doc/ot-userguide131/xhtml/index.html>, 2007. [WWW] [viitattu 29.06.2012].
- [42] P. Ranta, O. Nykänen, J. Salonen, S. Pohjolainen, ja K. Koskinen. Teollisuuden älykkäiden ja virtuaalisten konelaboratorioiden tuotantomenetelmien kehitys semanttisen kuvauksen avulla. tampereen teknillinen yliopisto. http://wiki.tut.fi/pub/SmartSimulators/Semogen/Semogen_loppuraportti_TTY_SmartSimulators_Final_211111.pdf, 2011. [WWW] [viitattu 28.06.2012] s. 205.
- [43] T. Renkema ja E. Berghout. Methodologies for information systems investment evaluation at the proposal stage: a comparative review. *Information and Software Technology*, 1997.
- [44] J. Salonen, O. Nykänen, P. Ranta, J. Nurmi, M. Helminen, M. Rokala, T. Palonen, V. Alarotu, K. Koskinen, ja S. Pohjolainen. An implementation of a semantic, web-based virtual machine laboratory prototyping environment. *The Semantic Web - ISWC*, 7032:221–236, 2011.
- [45] Jaakko Salonen. Wille visualisation system. tutwiki. <http://wiki.tut.fi/Wille/WebHome>, 2011. [WWW] [viitattu 11.12.2012].
- [46] E. Sandersa ja P. Stappersb. Co-creation and the new landscapes of design. *CoDesign: International Journal of CoCreation in Design and the Arts*, 4:5–18, 2008.
- [47] Dassault Systemes. 3dexperience universes for sustainable innovation. <http://www.3ds.com/>, 2013. [WWW] [viitattu 01.01.2013].

- [48] Dassault Systems. Solidworks. <http://www.solidworks.fi/>, 2013. [WWW] [viitattu 05.01.2013].
- [49] P. Tetlow, J. Pan, D. Oberle, E. Wallace, M Uschold, ja E. Kendall. Ontology driven architectures and potential uses of the semantic web in systems and software engineering. w3c, editors' draft. <http://www.w3.org/2001/sw/BestPractices/SE/ODA/>, 2006. [WWW] [viitattu 29.06.2012].
- [50] Tampere: Yhteiskuntatieteellinen tietoaarkisto. Kvalimotiv - menetelmäopetuksen tietovaranto. <http://www.fsd.uta.fi/menetelmaopetus/kvali/index.html>, 2006. [WWW] [viitattu 20.08.2012].
- [51] T. Tomiyama. Architecture-centric model-based product development. *The 13th Mechatronics Forum International Conference*, ss. 434–443, 2012.
- [52] Valtionvarainministeriö. Julkishallinnon xml-strategian toteuttaminen. http://www.vm.fi/vm/fi/04_julkaisut_ja_asiakirjat/01_julkaisut/04_hallinnon_kehittaminen/86189/86188_fi.pdf, 2004. [WWW] [viitattu 29.06.2012].
- [53] W3C. Owl 2 web ontology language document overview. owl working group, recommendation 27 october 2009. <http://www.w3.org/TR/owl2-overview/>, 2009. [WWW] [viitattu 14.07.2012].
- [54] W3C. Rich structured data markup for web documents. rdfa 1.1 primer, working group note 07 june 2012. <http://www.w3.org/TR/xhtml1-rdfa-primer/>, 2009. [WWW] [viitattu 16.07.2012].
- [55] W3C. Semantic web. <http://www.w3.org/standards/webarch/>, 2012. [WWW] [viitattu 14.07.2012].
- [56] P Wilson. *Computer Supported Cooperative Work: An Introduction*. Intellect Books, London, 1991. s. 116.
- [57] X3DOM. X3dom homepage. http://www.x3dom.org/?page_id=2, 2012. [WWW] [viitattu 24.12.2012].
- [58] A. Zoli ja S. Rush. *Cadd Primer: A General Guide to Computer Aided Design and Drafting-Cadd, CAD*. Mailmax Pub., Helsinki, 2000. s. 210.

A. LIITTEITÄ

A.1 Rinnakkaistettu simulaation kuuntelu WebSocket-yhteydellä

Rinnakkaisuuden mahdollistava JavaScript-koodi. Tämä funktio luo uuden säikeen simulaatiota kuuntelevalle JavaScript-ohjelmakoodille. Toteutushetkellä Worker-koodi toimi Google Chrome -selaimessa, mutta Firefox-selaimen vikatila sekoitti säikeen toiminnan. Tästä syystä selain ja sen Worker-tuki tarkastetaan, ja säikeistystä käytetään vain Google Chrome -selaimen kohdalla.

```
1 semolab.Simulation.prototype.WSUpdate = function(callback, responsiveness) {
2   //if workers are available and this is not a firefox browser
3   if (!!window.Worker && navigator.userAgent.indexOf("Firefox")== -1 ) {
4
5     function WorkerMessage(cmd, parameter) {
6       this.cmd = cmd;
7       this.parameter = parameter;
8     }
9
10    //create worker
11    var worker = new Worker('/static/lib/semolab/semolab.worker.js??');
12
13    worker.onerror = function(error) {
14      console.log("Worker error: " + error.message);
15    };
16
17    worker.onmessage = function (event) {
18      var message = event.data;
19      switch (message.cmd) {
20        case 'debug':
21          console.log(message.parameter);
22          break;
23        case 'simudata':
24          callback(message.parameter);
25          semolab.Simulation.update_count += 1;
26          break;
27        default:
28          if(semolab.debug) {
29            console.log("Error: Invalid data received from worker.")
30          }
31      }
32    };
33
34    //debug mode
35    if(semolab.debug) {
36      worker.postMessage(new WorkerMessage('debug', true));
37    }
38    //websocket API URL
39    worker.postMessage(
```

```

40     new WorkerMessage('url', semolab.api_urls['simulation_ws_get']) );
41     worker.postMessage(new WorkerMessage('pollTime', responsiveness) );
42     worker.postMessage(new WorkerMessage('start', null) );
43 }
44 // else we do it with one thread
45 else
46 {
47     websocket = new WebSocket(semolab.api_urls['simulation_ws_get']);
48     websocket.callback = callback;
49     websocket.onopen = function(evt) {
50         if(semolab.debug) {
51             console.log('websocket.open()');
52         }
53         // Start polling websocket
54         setInterval("pollWS()", responsiveness);
55     };
56     websocket.onclose = function(evt) {
57         if(semolab.debug) {
58             console.log('websocket.close()');
59         }
60     };
61     websocket.onmessage = function(evt) {
62         obj = JSON.parse(evt.data); // Parse JSON data
63         semolab.Simulation.update_count += 1;
64         this.callback(obj);
65     };
66     websocket.onerror = function(evt) {
67         if(semolab.debug) {
68             console.log('websocket.error()');
69         }
70     };
71 }
72 }

```

Tämä on omassa säikeessään eli moniydinprosessorissa omalla ytimellään ajossa oleva koodi. Edellinen funktio kutsuu tämän ohjelmakoodin taustalle ajoon. Säie avaa WebSocket-yhteyden Semoplayer-palvelimeen ja alkaa välittämään simulaation tilaa palvelimelta VML:n näyttimille.

```

1 var debug = false;
2 var websocketURL = "";
3 var pollTime = 25;
4
5 function WorkerMessage(cmd, parameter) {
6     this.cmd = cmd;
7     this.parameter = parameter;
8 }
9

```

```
10 function pollWS() {
11     websocket.send('{}'); // Send empty JSON variable
12 }
13
14 onmessage = function(event)
15 {
16     // Accessing to the message data sent by the main page
17     var messageSent = event.data;
18
19     // Testing the command sent by the main page
20     switch (messageSent.cmd) {
21         case 'debug':
22             debug = messageSent.parameter;
23             break;
24         case 'url':
25             websocketURL = messageSent.parameter;
26             break;
27         case 'pollTime':
28             pollTime = messageSent.parameter;
29             break;
30         case 'start':
31             openSocket();
32             break;
33     }
34 }
35
36 function openSocket()
37 {
38     websocket = new WebSocket(websocketURL);
39
40     websocket.onopen = function(evt) {
41         if(debug) {
42             postMessage(new WorkerMessage('debug', 'websocket.open()'));
43         }
44         // Start polling websocket
45         setInterval("pollWS()", pollTime);
46     };
47
48     websocket.onclose = function(evt) {
49         if(debug) {
50             postMessage(new WorkerMessage('debug', 'websocket.close()'));
51         }
52     };
53
54     websocket.onmessage = function(evt) {
55         obj = JSON.parse(evt.data); // Parse JSON data
56         postMessage(new WorkerMessage('simudata', obj));
```

```
57     };
58
59     websocket.onerror = function(evt) {
60         if(debug) {
61             postMessage(new WorkerMessage('debug', 'websocket.error()'));
62         }
63     };
64 }
```

A.2 Toimintomallin semanttinen kuvaus

Toiminnot mahdollistavat toisia toimintoja (enables) ja muodostavat toimintojen verkon. Toimintojen verkossa kuljetut polut muodostavat lopulta toimintoketjun. Toimintoon liittyvä komponentti vaikuttaa tilallaan siihen, mitkä ovat toiminnon laukaisemia toimintoja.

- Toiminto on komponentin tai kokoonpanon suorittama tapahtuma.
- Toiminto riippuu komponentin tilasta ja voi muuttaa komponentin tilaa.
- Tilat kuvaavat komponentin mahdollisuutta suorittaa toiminto.
- Toiminto mahdollistaa toisia toimintoja, jotka toteutuvat niihin liittyvien komponenttien tilojen mukaan.
- Toimintoketju on yhden tai useamman toiminnon kokonaisuus, joka alkaa ja päättyy tietyistä toiminnoista.
- Toimintoketjussa yksittäistä toimintoa voidaan kutsua askeleeksi (step).
- StartStep on toiminnosta periytetty askel, joka aloittaa tietyn toimintoketjun.
- Sama askel voi olla osa useaa eri toimintoketjua.
- Toimintoketju voi olla osittain tai kokonaan osa toista toimintoketjua.
- Ensimmäinen askel laukaisee lisää askelia komponenttien tilojen perusteella.
- Kun toimintoketju on kuljettu loppuun, niin voidaan tarkastella sen lopputulosta.

Validi RDF/XML- ja OWL/XML-kuvaus toimintoketjusta. Samassa tiedostossa on OWL-ontologiamalli toimintoketjulle sekä esimerkkinä yhden toimintoketjun kuvaus RDF-muodossa.

```
1 <?xml version="1.0"?>
2
3 <!DOCTYPE rdf:RDF [
4   <!ENTITY dc "http://purl.org/dc/elements/1.1/" >
5   <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
6   <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
7   <!ENTITY semogen "http://www.tut.fi/projects/semogen/" >
8   <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
9   <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
10 ]>
11
```



```
12 <rdf:RDF
13   xmlns="http://www.tut.fi/projects/semogen/functionchain.owl#"
14   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
15   xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
16   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
17   xmlns:owl="http://www.w3.org/2002/07/owl#"
18   xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
19   xmlns:semogen="http://www.tut.fi/projects/semogen/"
20   xml:base="http://www.tut.fi/projects/semogen/functionchain.owl#">
21
22   <owl:Class rdf:about="&semogen;Component">
23     <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
24       Component
25     </rdfs:label>
26   </owl:Class>
27
28   <owl:ObjectProperty rdf:ID="FunctionOf">
29     <rdfs:range rdf:resource="&semogen;Component" />
30     <rdfs:domain rdf:resource="#Step" />
31     <owl:inverseOf>
32       <owl:ObjectProperty rdf:ID="hasFunction" />
33     </owl:inverseOf>
34     <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
35       FunctionOf
36     </rdfs:label>
37   </owl:ObjectProperty>
38
39   <owl:Class rdf:ID="Step">
40     <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
41       Step
42     </rdfs:label>
43   </owl:Class>
44
45   <owl:Class rdf:ID="StartStep">
46     <rdfs:subClassOf rdf:resource="#Step" />
47   </owl:Class>
48
49   <owl:ObjectProperty rdf:ID="dependsOn">
50     <owl:inverseOf>
51       <owl:ObjectProperty rdf:ID="enables" />
52     </owl:inverseOf>
53     <rdfs:subPropertyOf>
54       <owl:ObjectProperty rdf:ID="neighborStep" />
55     </rdfs:subPropertyOf>
56     <rdfs:subPropertyOf>
57       <owl:TransitiveProperty rdf:ID="hasPrerequisite" />
58     </rdfs:subPropertyOf>
```

```
59   <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
60     depends on
61   </rdfs:label>
62 </owl:ObjectProperty>
63
64 <owl:ObjectProperty rdf:ID="otherStep">
65   <rdfs:range rdf:resource="#Step"/>
66   <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
67     other step
68   </rdfs:label>
69   <rdfs:domain rdf:resource="#Step"/>
70 </owl:ObjectProperty>
71
72 <owl:ObjectProperty rdf:about="#enables">
73   <rdfs:subPropertyOf>
74     <owl:ObjectProperty rdf:about="#neighborStep"/>
75   </rdfs:subPropertyOf>
76   <rdfs:subPropertyOf>
77     <owl:TransitiveProperty rdf:ID="prerequisiteFor"/>
78   </rdfs:subPropertyOf>
79   <rdfs:range rdf:resource="#Step"/>
80   <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
81     enables
82   </rdfs:label>
83   <rdfs:domain rdf:resource="#Step"/>
84 </owl:ObjectProperty>
85
86 <owl:ObjectProperty rdf:about="#neighborStep">
87   <rdfs:subPropertyOf>
88     <owl:TransitiveProperty rdf:ID="inSameChain"/>
89   </rdfs:subPropertyOf>
90   <rdfs:range rdf:resource="#Step"/>
91   <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
92     neighbor step
93   </rdfs:label>
94   <rdfs:domain rdf:resource="#Step"/>
95 </owl:ObjectProperty>
96
97 <owl:TransitiveProperty rdf:about="#prerequisiteFor">
98   <rdfs:subPropertyOf rdf:resource="#otherStep"/>
99   <rdfs:range rdf:resource="#Step"/>
100  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
101    prerequisite for
102  </rdfs:label>
103  <rdfs:domain rdf:resource="#Step"/>
104 </owl:TransitiveProperty>
105
```

```

106 <owl:TransitiveProperty rdf:about="#hasPrerequisite">
107   <rdfs:subPropertyOf rdf:resource="#otherStep"/>
108   <rdfs:range rdf:resource="#Step"/>
109   <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
110     has prerequisite
111 </rdfs:label>
112   <rdfs:domain rdf:resource="#Step"/>
113 </owl:TransitiveProperty>
114
115 <owl:TransitiveProperty rdf:about="#inSameChain">
116   <rdfs:range rdf:resource="#Step"/>
117   <rdfs:domain rdf:resource="#Step"/>
118 </owl:TransitiveProperty>
119
120 <owl:AnnotationProperty
121   rdf:about="http://www.w3.org/2000/01/rdf-schema#label"/>
122 <owl:AnnotationProperty
123   rdf:about="http://www.w3.org/2002/07/owl#versionInfo"/>
124
125 <StartStep rdf:ID="OhjauksenAsentoX">
126   <FunctionOf rdf:resource="#CAN_B_LIFT_V"/>
127   <enables>
128     <Step rdf:ID="Ohjausventtiilin_Tila_Muuttuu">
129       <FunctionOf rdf:resource="#B_lift_4_3_control_valve"/>
130       <enables>
131         <Step rdf:ID="Vasen_Solenoidi_Virroittuu">
132           <FunctionOf rdf:resource="#Y26"/>
133           <enables>
134             <Step rdf:ID="Venttiilin_Kara_Liikkuu">
135               <FunctionOf rdf:resource="#B_lift_4_3_control_valve"/>
136               <enables rdf:resource="#Portti_P_kytkeytyy_Porttiin_A" />
137               <enables rdf:resource="#Portti_T_kytkeytyy_Porttiin_B" />
138               <rdfs:label
139                 rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
140                 Venttiilin kara liikkuu
141               </rdfs:label>
142             </Step>
143           </enables>
144           <rdfs:label
145             rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
146             Vasen solenoidi virroittuu
147           </rdfs:label>
148         </Step>
149       </enables>
150     </rdfs:label
151     rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
152     Ohjausventtiilin tila muuttuu

```

```
153     </rdfs:label>
154 </Step>
155 </enables>
156 <rdfs:label
157   rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
158   Ohjauksen asento X
159 </rdfs:label>
160 </StartStep>
161
162 <Step rdf:ID="Portti_P_kytkeytyy_Porttiin_A">
163   <FunctionOf rdf:resource="#Port_P"/>
164   <FunctionOf rdf:resource="#Port_A"/>
165   <enables
166     rdf:resource="#Paine_Nousee_Kuormaa_Vastaavaksi_Linjassa" />
167 <enables
168   rdf:resource="#Sylinterin_kuormitus_on_liian_suuri" />
169 <rdfs:label
170   rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
171   Portti P kytkeytyy porttiin A
172 </rdfs:label>
173 </Step>
174
175 <Step rdf:ID="Portti_T_kytkeytyy_Porttiin_B">
176   <FunctionOf rdf:resource="#Port_T"/>
177   <FunctionOf rdf:resource="#Port_B"/>
178   <enables
179     rdf:resource="#Paine_Nousee_Kuormaa_Vastaavaksi_Linjassa" />
180 <enables rdf:resource="#Sylinterin_kuormitus_on_liian_suuri" />
181 <rdfs:label
182   rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
183   Portti T kytkeytyy porttiin B
184 </rdfs:label>
185 </Step>
186
187 <Step rdf:ID="Paine_Nousee_Kuormaa_Vastaavaksi_Linjassa">
188   <FunctionOf rdf:resource="#B_lift_4_3_control_valve"/>
189   <enables>
190     <Step rdf:ID="Paine_Nousee_Kuormaa_Vastaavaksi_Kammiossa">
191       <FunctionOf rdf:resource="#A_lift_cylinder" />
192       <enables>
193         <Step rdf:ID="Sylinteri_Liikkuu">
194           <FunctionOf rdf:resource="#A_lift_cylinder" />
195           <rdfs:label
196             rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
197             Sylinteri liikkuu
198           </rdfs:label>
199         </Step>
```

```
200     </enables>
201     <rdfs:label
202       rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
203       Paine nousee kuormaa vastaavalle tasolle kammiossa
204     </rdfs:label>
205   </Step>
206 </enables>
207 <rdfs:label
208   rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
209   Paine nousee kuormaa vastaavalle tasolle linjassa
210 </rdfs:label>
211 </Step>
212
213 <Step rdf:ID="Sylinterin_kuormitus_on_liian_suuri">
214   <FunctionOf rdf:resource="#port-2_21"/>
215   <enables>
216     <Step rdf:ID="Ei_virtausta_ja_ei_liiku">
217       <rdfs:label
218         rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
219         Ei virtausta; Sylinteri ei liiku
220       </rdfs:label>
221     </Step>
222   </enables>
223   <rdfs:label
224     rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
225     Sylinterin kuormitus on liian suuri
226   </rdfs:label>
227 </Step>
228
229 </rdf:RDF>
```

A.3 Semanttinen datalehti

Semanttisella datalehdellä on sekä koneluettava semanttinen malli A.1, että ihmisluettava näkymä A.2.

View
Edit
Discussion
Describe
History

[🏠](#) » [exampleproject](#) » [semogen](#) » [datasheet](#) » [sheets](#) » [vdpumps](#) » PO1234

po1234 Resource

po1234

```

calli:administrator /group/admin (describe)
calli:editor /group/staff (describe)
calli:reader /group/users (describe)
dc:date 2012-09-04^^xsd:date
dc:description Medium duty pump has been developed to cover the mobile
as well as the industrial market when it comes to medium duty
applications. It has a 3D CAD based, computer optimized design
that leads to a considerable reduction of noise. Due to minimized
size and weight as well as an adapted compensator design
installation is very easy. The pump shows high overall efficiency
and wide operating temperature capability.
default1:image http://matriisi.ee.tut.fi/~nurmi28/datasheet/parkerpump.png^^xsd:anyURI
default1:price 1000^^xsd:integer
hdvdp:circuit open
hdvdp:controlDevices PRESSURE LIMITER CONTROL WITH ELECTRONIC ADJUSTMENT
hdvdp:directionOfRotation CCW/CW
hdvdp:displacement 50^^xsd:integer
hdvdp:electricalConnection HIRSCHMANN DIN EN 175 301-803-A
hdvdp:flowAt1500 50^^xsd:integer
hdvdp:fluidHFA HFA
hdvdp:fluidHFC HFC
hdvdp:fluidOil oil
hdvdp:fluidWater water
hdvdp:manufacturer PumpManuFac
hdvdp:maxFlow 140^^xsd:integer
hdvdp:maxSpeed 2000^^xsd:integer
hdvdp:mechanicalEfficiency [0 1.5 1.8 1.9 2 1.5 1.2]
hdvdp:mounting DIN 2-hole
hdvdp:nominalPressure 210^^xsd:integer
hdvdp:overallEfficiency [0 0.7 0.8 0.9 1 0.9 0.8]
hdvdp:peakPressure 300^^xsd:integer
hdvdp:pressureVector [0 5 10 15 20 25 30]
hdvdp:saftEnd SAE J744
hdvdp:viscosityAreaLower 5^^xsd:integer
hdvdp:viscosityAreaUpper 25^^xsd:integer
hdvdp:volumetricEfficiency [0 0.4 0.5 0.6 0.6 0.4]
hdvdp:weight 20^^xsd:integer
prov:wasGeneratedBy /activity/2012/09/28/t13a085c3857x2748 (describe)
rdf:type /exampleproject/semogen/datasheet/Mechatronics_datasheet_-_for_hydraulics_-_
_variable_displacement_pump (describe)
rdfs:label PO1234
          
```

As RDF/XML

As Turtle

Kuva A.1: Semanttisen datalehden koneluettava semanttinen tieto.


View Edit Discussion Describe History

» exampleproject » semogen » datasheet » sheets » vdpumps » PO1234

PO1234

PumpManuFac 1000 euros

Medium duty pump has been developed to cover the mobile as well as the industrial market when it comes to medium duty applications. It has a 3D CAD based, computer optimized design that leads to a considerable reduction of noise. Due to minimized size and weight as well as an adapted compensator design installation is very easy. The pump shows high overall efficiency and wide operating temperature capability.



Resources

- [homepage](#)
- [symbol](#)
- [data](#)
- [mfile](#)

Technical information

PRESSURE LIMITER CONTROL WITH ELECTRONIC ADJUSTMENT

Circuit	Displacement	Nominal Pressure	Peak Pressure
open	50	210	300
Max Speed	Max Flow	Flow at 1500	
2000	140	50	
Viscosity area - lower	Viscosity area - upper		
5	25		
Weight	Direction of Rotation		
20	CCW/CW		
Shaft End	Mounting	Electrical Connection	
SAE J744	DIN 2-hole	HIRSCHMANN DIN EN 175 301-803-A	

Fluids:

- water
- oil
- HFA
- HFC

Pressure-Efficiency

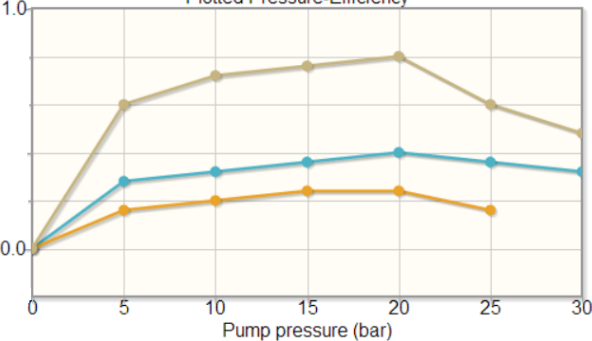
[0 5 10 15 20 25 30]

[0 0.4 0.5 0.6 0.6 0.4]

[0 1.5 1.8 1.9 2 1.5 1.2]

[0 0.7 0.8 0.9 1 0.9 0.8]

Plotted Pressure-Efficiency



PDF file

Kuva A.2: Semanttisen datalehden katselunäkymä ihmisille.