KARELIA-AMMATTIKORKEAKOULU

Tietotekniikan koulutusohjelma

Sami Honkanen

BPM JA KÄYTTÖLIITTYMÄT

Opinnäytetyö

Joulukuu 2013

Karelia AMMATTIKORKEAKOULU	**OPINNÄYTETYÖ** **Joulukuu 2013** **Tietotekniikan koulutusohjelma** Karjalankatu 3 80200 JOENSUU

Tekijä
Sami Honkanen

Nimeke
BPM ja käyttöliittymät

Toimeksiantaja
Karelia-ammattikorkeakoulu

Tiivistelmä

Opinnäytetyön tavoitteena oli selvittää, miten Business Process Model & Notation -standardin mukaisissa liiketoimintaprosessikuvauksissa mallinnetaan käyttäjien tehtäviä ja miten ne implementoidaan liiketoiminnanohjaus ohjelmistoihin ja -käyttöliittymiin. Opinnäytetyössä käydään asteittain läpi liiketoimintaprosessin laatiminen käyttäjätehtäville yhdessä ohjelmistossa ja miten liiketoimintaprosessien hallinta on kehittynyt sekä työn kannalta tärkeimmät tekniikat.

Tutkimus rajattiin yhteen käytetyimmistä liiketoimintaprosessin hallintaohjelmistoista, Intalio | bpms:iin mistä on saatavilla ilmainen yhteisöversio. Tälle ohjelmistolle laadittiin notaation mukaiset liiketoimintaprosessit, mitkä käyvät läpi kaikki mahdolliset tavat käyttäjätehtäville.

Työn tuloksena saatiin selville, että notaation mukaiset tapaukset eivät välttämättä siirry ohjelmistosta toiseen, ohjelmistoissa tapahtuvien liiketoimintaprosessikuvausten käsittelyjen seurauksena. Tuloksena syntynyttä dokumenttia voidaan toteutuksen osalta käyttää opetusmateriaalin pohjana. Mahdollisia jatkokehityksen suuntauksia olisivat notaation testaus muilla ohjelmistoilla ja prosessien tehokkuuden mittaaminen.

Kieli suomi	Sivuja 32 Liitteet 1 Liitesivumäärä 1

Asiasanat
bpmn, bpm, liiketoimintaprosessi, toteutettavuustutkimus

THESIS
December 2013
Degree Programme in Information Technology
Karjalankatu 3
FI 80200 JOENSUU
FINLAND

Author
Sami Honkanen

Title
BPM and user interfaces

Commissioned by
Karelia University of Applied Sciences

Abstract

The aim was to find out how user tasks are modeled in business process descriptions, using Business Process Modeling Notation standard, and how they are implemented into business operations management software and user interfaces. The thesis goes through step by step of drawing up a process for user tasks in software and how the business process management has evolved, as well as the main technologies used in the work.

The study was limited to one widely used business process management software, Intalio | BPMS which is available in a free community version. For this software were drawn business processes, in accordance with the Business Process Modeling Notation, which go through all the possible ways of user tasks.

As a result, it was found out that the notation do not necessarily move from the software to another because of the way the software handles the notation. Implementation part of the resulting document can be used as a base for teaching material. The potential for further development trends would be testing the notation with other software and measurement of the effectiveness of processes.

Language
Finnish

Pages 32
Appendices 1
Pages of Appendices 1

Keywords
bpmn, bpm, business process, feasibility study

Sisältö

1 Johdanto 5
2 Tutkimuksen tarkoitus 6
3 BPM:n kehitys 6
4 Web-tekniikoita 8
4.1 SOA 8
4.2 HTTP 9
4.3 URI 9
4.4 XML 9
4.5 Web Service 11
4.6 SOAP 11
4.7 WSDL 12
5 Liiketoimintaprosesseihin liittyvät tekniikat 13
5.1 BPM 13
5.2 BPMN 14
5.3 BPEL 15
5.4 Apache ODE 15
5.5 Intalio | bpms 15
6 Toteutus 17
6.1 BPMN ja User Task 17
6.2 Liiketoimintaprosessin implementointi 19
6.2.1 Liiketoimintaprosessin kuvaus ohjelmointiympäristöllä 19
6.2.2 Liiketoimintaprosessin julkaisu 21
6.3 Laajennettu prosessi 25
6.4 Intalio | Ajax 28
7 Tulokset 32
8 Pohdinta 33
Lähteet 34

Liitteet

Liite 1 Lyhenteet

1 Johdanto

Aiemmin liiketoiminta rakentui liiketoiminnallisen prosessin ympärille, esimerkiksi auton kokoonpanolinja. Ajan saatossa tästä ajattelusta on päästy eroon ja liiketoiminta on muuttunut paljon joustavammaksi prosessimuotoisen ajattelun sijasta liiketoiminnalliseen ajatteluun, missä prosessi tukee liiketoimintaa eikä toisinpäin.

Liiketoiminnallisen prosessin tarkoituksena on suorittaa jokin liiketoiminnalle tärkeä asia kuten tietyn tuotteen valmistus. Liiketoimintaprosessia kuvaamaan on kehittynyt oma liiketoimintaprosessinkuvausstandardi BPMN (Business Process Model & Notation) [1]. Se on kehitetty eritoten sitä varten että liiketoiminnan kaupallinen ja tekninen henkilöstö voivat helposti kommunikoida keskenään molemmille ymmärrettävällä kielellä. Sen avulla yrityksen johto voi helposti piirtää liiketoimintaprosessin minkä tekninen henkilöstö voi muuttaa toimivaksi järjestelmäksi.

Tavoite työssä oli toteutettavuustutkimus siitä miten liiketoimintaprosessissa mallinnetaan käyttäjän antamat ja koneelliset syötteet. Työn tarkoituksena on selvittää, miten liiketoimintaprosessin kuvausstandardin mukaan mallinnetut käyttäjän antamat syötteet, vastaanotetaan prosessimoottorilla ja laatia ohjeet miten tutkimuksessa selvitetyt tapaukset voidaan jäljentää liiketoimintaprosessiin. Työssä selvitetään kaikki eri tapaukset miten syötteitä voidaan liiketoimintaprosessiin teknisen henkilöstön näkökulmasta liittää. Työ suoritettiin Intalio | bpms -ohjelmistolla, mutta sen tuloksia pitää pystyä soveltamaan myös muihin ohjelmistoihin jotka käyttävät liiketoimintaprosessinotaatiota.

Työ tehtiin avoimen lähdekoodin sekä ilmaisessa jakelussa olevilla ohjelmilla, joista osasta on saatavilla maksullisia versioita lisäominaisuuksilla. Raportoinnissa käytettiin myös maksullisia sovelluksia. Pohjana työssä käytettiin Jussi-Pekka Turusen (2013) opinnäytetyötä sekä teoksia liittyen palvelukeskeiseen arkkitehtuuriin, web-ohjelmointiin ja BPM:iin.

2 Tutkimuksen tarkoitus

Työn tarkoituksena oli selvittää miten BPMN-notaatiolla muodostetussa liiketoimintaprosesseissa käyttäjän suorittamat tehtävät ilmentyvät ja mitä erilaisia tapoja niiden implementointiin Intalio | bpms ohjelmistossa on tarjolla. Tavoitteena oli saada selvitys eri tavoista miten käyttäjän antamat komennot saadaan lähetettyä prosessiin, tuottaa niistä mahdolliset esimerkki tapaukset ja mahdollisesti tuottaa ohjeistusta niiden implementointiin.

Suunnitteluvaiheessa lähdin jakamaan aihetta pienempiin osiin aiheen koon hahmottamiseksi. Tässä päädyin kolmeen pääkysymykseen:

- Mikä BPMN on ja mikä sen tarkoitus on?
- Mitä ovat käyttäjättehtävät (User Task)?
- Mikä on Intalio | bpms?

Näiden pohjalta aiheeseen lähdettiin tutustumaan tarkemmin.

3 BPM:n kehitys

Liiketoimintaprosessit ovat jatkumoa teolliselle vallankumoukselle mikä alkoi 1800-luvun lopussa. Tästä hyvänä esimerkkinä on Henry Ford ja hänen tekemänsä valmistusprosessi autoille vuonna 1903. [2, s. 1.]

Liiketoimintaprosessien systemaattinen kehitys alkoi 1900-luvun alussa Frederic Winslow Taylorin julkaiseman kirjan Principles of Scientific Management vauhdittamana [2, s. 2.]. Hän korosti yksinkertaisuutta ja sitä että prosessia tutkimalla ja mittaamalla sitä voitaisiin kehittää tuotannon parantamiseksi. Hänellä oli neljä sääntöä tämän toteuttamiseen: ”Korvaa sinnepäin menettelyt tieteellisesti todistetuilla tavoilla. Valitse ja kouluta henkilökunta, äläkä anna heidän opettaa itseään. Anna tarkat ohjeet työn tekemiseen ja valvo työtä. Jaa työ tasaisesti esimiesten ja työntekijöiden välillä.”. [3, s. 56.]

Tämänhetkisen kokonaisvaltaisen liiketoimintaprosessin tyyli perustuu vasta Michael Porterin tekemään pohjustustyöhön hänen julkaisemassaan kirjassa (1985) [4]. Siinä tuodaan esiin hänen konseptinsa arvoketjusta (kuva 1). Liittämällä arvoketju liiketoimintaprosessia kuvaavaan vuokaavioon saadaan aikaan yksi prosessin kulkua kuvaava kaavio. Geary Rummler käytti tämän tyylistä kaaviota ensimmäisen kerran jo vuonna 1984. Rummler edisti Alan Brache:n kanssa sitä miten prosesseja analysoidaan, ja kehitään. Suurimman edistyksen Rummler toi aikaisempaan kaavion siinä että hän otti huomioon mitä ongelmia osastojen välillä tapahtuvassa viestinnässä oli. Hän väitti että vain hahmottamalla koko prosessi voitaisiin näistä ongelmista päästä eroon. Vaikka Alan Brache ja Geary Rummler kirjoittivat yhdessä kirjan aiheesta, eivät he käynnistäneet uudistamisprosessia yrityksissä vaan sen tekivät Michael Hammer ja Thomas H. Davenport. He saivat tämän aikaan 1990-luvulla julkaisemistaan kirjoista, mitkä sisälsivät esimerkkejä yritysten kasvavista suorituskyvyistä. [2, s. 3–6.]

Kuva 1. Porterin arvoketju.

Prosessin kehityksen lisäksi kehittyivät laadunhallintaan keskittyneet liikkeet. 1980-luvun lopulla TQM Total Quality Management oli hyvin suosittu. Samoihin aikoihin kehittyi Rummler-Brache suuntauksen ja laadunhallintaliikkeen vuorovaikutuksena Six Sigma -liike. Se kehittyi Motorolalla 1980-luvulla ensimmäisenä Bill Smithin formuloimana Rummlerin konsultoinnin vaikutuksesta. Six Sigmassa painotetaan jatkuvaa prosessin laadun valvontaa ja sen kehittämistä. Tarkoituksena Six Sigmassa on saada tuotteen virheet mahdollisimman pieneksi niin että jopa 99,99966% tuotteista olisivat

virheettömiä. Sen levitessä Motorolalta muille yrityksille siitä kehittyi kattava valmennusohjelma koko yrityshenkilöstölle. [2, s. 8; 5, s. 3–5.]

Six Sigman ohella hieman myöhemmin 1990-luvulla kehittyi Lean liike [1, s 18]. Leanissa tähdätään siihen että kaikki mahdollinen turha jäte poistetaan prosesseista, esimerkiksi liian suuret varastot. Nämä suuntaukset yhdistyivät vuosituhannen vaihteessa Lean Six Sigmassa mikä otti huomioon Six Sigman laadun ja Leanin suorituskyvyn parantamiset [2, s. 18; 4, s. 2–3.].

Yhdysvaltojen puolustusvoimissa kehitettiin 1990-luvulla CMM (Capability Maturity Model), mikä jakaa prosessit viiteen kehitys asteeseen. Se oli aluksi tarkoitettu ohjelmistojen laadun määrittelyyn. Tästä edelleen kehittyi 2000-luvulla liiketoimintaprosesseja vastaava kypsyysmalli BPMM. [2, s. xxxiii, 18.]

Näitten yhteisvaikutuksen tuloksena syntyi tämänhetkinen BPM (Business Process Management), missä yhdistyvät liiketoimintaprosessi, laadunhallinta, optimointi ja mittaus.

4 Web-tekniikoita

Tässä osassa käydään läpi työn kannalta merkitsevimpiä verkossa käytettyjä tekniikoita. Tekniikoihin tutustutaan ensin suuremman mittakaavan asiaan ja siten pienempiin osuuksiin. Näitä tekniikoita sovelletaan työn toteutusosassa joten niihin on syytä tutustua.

4.1 SOA

SOA (service oriented architecture) on ohjelmisto suunnittelun menetelmä missä suuri tai monimutkainen ohjelmisto muodostuu useista pienemmistä ohjelmisto ryhmistä eli palveluista. Jokainen palvelu, mistä suurempi kokonaisuus rakentuu, on yksittäinen ohjelmisto moduuli mikä itsessään ei tee mitään vaan suorittaa siihen määriteltyjä funktioita. Tämän johdosta jokainen moduuli on uudelleen käytettävä useissa eri kokonaisuuksissa muuttamalla sitä miten yksittäinen palvelu tuotosta käsittelee.

SOA yleensä tarjoaa tien palveluiden käyttäjille mitä palveluita on saatavilla. Sillä, millä kielellä palveluita luodaan, ei yleensä ole väliä hyvin laadittujen rajapintojen takia. Keskustelu palveluihin tapahtuu palvelulle määritetyllä tavalla esimerkiksi XML:llä tai JSON:lla. Näistä JSON:in käyttö on noussut ajan saatossa tänä päivänä käytetyimmäksi formaatiksi.

4.2 HTTP

HTTP (Hypertext Transfer Protocol) on sovellustason protokolla hajautetuille yhteistyössä toimiville hypermedia tietojärjestelmille. Se on pyyntö/vastaus protokolla. Asiakas lähettää pyynnön palvelimelle, mikä sisältää komennon, osoitteen ja protokollan version. Tätä seuraa MIME-tyyppinen viesti sisältäen pyyntöön liittyvät tarkemmat tiedot. Palvelin vastaa tähän pyyntöön ilmoittamalla pyynnön tilan ja MIME-tyyppisen viestin, mikä sisältää mahdollisesti pyydetyt tiedot. [6, s. 7, 12.]

4.3 URI

URI (Uniform Resource Identifier) toiselta nimeltään URL on jonkin resurssin yksilöllinen osoitin. Se on ainut verkossa nimeämiseen tarkoitettu tekniikka jolla verkossa olevia resursseja voidaan käyttää. [7, s. 1–2.]

4.4 XML

XML (Extensible Markup Language) on kieli millä kuvataan tietoa. Se on helposti ymmärrettävää sekä ihmisille että koneille. XML-tiedostolla on hyvin looginen hierarkkinen rakenne sen koostuessa yksiköistä (kuva 2). XML on W3C:n (World Wide Web Consortium) kehittämä ja se on tarkoitettu eritoten yksinkertaiseen tiedonsiirtoon Internetissä. [8]

Yksi XML tavoitteista on erottaa tieto ja kuinka tieto esitetään. Tämä erotus on helposti toteutettu XML:ssä sillä siitä puuttuvat valmiit tyylittelyominaisuudet kokonaan. XML tiedostojen tyylittelyyn on olemassa erillinen XSLT (Extensible Stylesheet Language Transformations) merkintäkieli. [9, s. 10.]

XML-tiedostoille on olemassa useita tapoja millä niitä voidaan kuvailla mutta käytetyin on luultavasti XML Schema. XML-skeeman avulla voidaan kuvata XML-tiedoston rakenne ja rajoitteet. Skeeman avulla voidaan todeta myös XML-tiedoston oikeellisuus. Kuva 2 osoittaa esimerkin XML-tiedostosta jonka täytyy noudattaa kuvassa 3 olevaa skeemaa. [9, s. 10–12.]

```
<?xml version="1.0" encoding="UTF-8"?>
<exampleroot documentelement="example"
             xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
             xsi:noNamespaceSchemaLocation="newXmlSchema.xsd">
    <example1 name="exampleattribute" >example1</example1>
    <example2>
        <example3>example2</example3>
        <example4>example3</example4>
    </example2>
</exampleroot>
```

Kuva 2. XML-tiedosto.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
             elementFormDefault="qualified">
    <xsd:element name="exampleroot">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="example1" type="example1"/>
                <xsd:element name="example2" type="example2"/>
            </xsd:sequence>
            <xsd:attribute name="documentelement" />
        </xsd:complexType>
    </xsd:element>
    <xsd:complexType name="example1">
        <xsd:simpleContent>
            <xsd:extension base="xsd:string">
                <xsd:attribute name="name" type="xsd:string" />
            </xsd:extension>
        </xsd:simpleContent>
    </xsd:complexType>
    <xsd:complexType name="example2">
        <xsd:sequence>
            <xsd:element name="example3" type="xsd:string" />
            <xsd:element name="example4" type="xsd:string" />
        </xsd:sequence>
    </xsd:complexType>
</xsd:schema>
```

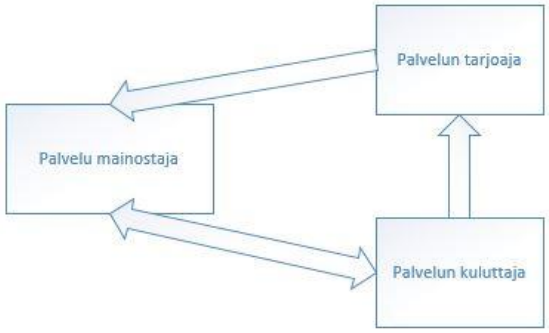
Kuva 3. XML Schema.

4.5 Web Service

W3C:n määritelmän mukaan Web Service on ohjelma joka pystytään tunnistamaan URI:lla ja jonka liitynnät ja sidokset pystytään määrittelemään, kuvaamaan ja tunnistamaan XML artefakteilla. Ohjelmiston pitäisi pystyä myös viestimään muiden ohjelmien kanssa XML-pohjaisilla viesteillä Internet-pohjaisten protokollien avulla. [10]

Web servicen tehtävä on yleensä isomman kokonaisuuden yksittäinen laskentayksikkö minkä tarkoituksena on suorittaa yksi toiminto. Ohjelmistokehittäjien tavoitteena on saada yhdistetty mahdollisimman monta palvelua suuremmaksi kokonaisuudeksi jonka avulla voidaan suorittaa monimutkaisempia toimintoja. Koska niitä voidaan käyttää Internetin välityksellä mahdollistavat ne houkuttelevampia ja skaalaantuvampia ohjelmistoarkkitehtuureja. [11, s. 21.]

Kuva 4 osoittaa yleisimmän tavan miten palveluita käytetään. Palvelun tarjoaja mainostaa palvelujaan jonkinlaisen mainostajan avulla kuten WSDL-tiedoston. Palvelun kuluttaja taas lukee mainostajan mainostaman palvelun joka kertoo missä palvelu sijaitsee ja mitä operaatioita se tarjoaa. Näiden tietojen avulla palvelun kuluttaja voi käyttää palvelun tarjoajan palveluita.



Kuva 4. Web Service malli.

4.6 SOAP

SOAP on kevyt tapa tiedonsiirtoon hajautetussa ympäristössä. Sen tärkeimpiä ominaisuuksia ovat yksinkertaisuus ja laajennettavuus. Yksinkertaisuudesta tosin ei olla aivan

yhtä mieltä 1.2 version julkistamisen jälkeen. Se on XML-pohjainen viestinvälitys kehys jota voidaan käyttää millä tahansa ohjelmalla tai protokollalla. Alun perin SOAP oli protokolla mikä käytti HTTP-protokollaa tiedonsiirtoon mutta myöhemmin se on määritelty tarkemmin. Vaikka SOAPia voidaan käyttää muillakin tiedonsiirtoprotokollilla käytännössä lähes kaikki käyttävät http-protokollaa. [12]

SOAP-viesti on hyvin järjestelty XML dokumentti kuten kuvan 5 mukainen SOAP-viestin pohja. Se koostuu neljästä osasta: Envelopesta, Headeristä, Bodystä ja Faultista. Välttämättömiä osia viestistä ovat vain Envelope ja Body. Envelope kertoo että dokumentti on SOAP-viesti ja Body sisältää siirrettävän tiedon. Header ja Fault osat ovat valinnaisia. Header osaa käytetään yleisesti viestiin liittyvään metadataan ja Fault on viestin prosessoinnissa tapahtuvien virheiden tutkimiseen. [12]

```
1  <env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
2   <env:Header>
3    <env:Fault>
4    </env:Fault>
5   </env:Header>
6   <env:Body>
7    <env:Fault>
8    </env:Fault>
9   </env:Body>
10 </env:Envelope>
```

Kuva 5. Esimerkki SOAP viestistä.

4.7 WSDL

WSDL (Web Service Description Language) on XML-pohjainen kieli mikä antaa mallin kuinka verkkopalveluita kuvataan [13, s. 16.]. WSDL tiedostossa kuvataan abstraktisti mitä operaatioita ja viestejä palvelu sisältää ja ne kootaan yhteen päätepisteeseen. Päätepisteeseen yhdistämällä palvelun kuluttaja saa WSDL tiedoston jonka avulla selviää mitä operaatioita palvelussa on käytössä.

WSDL:stä on käytössä kahta versiota 1.1 ja 2.0. Kuitenkin suurin osa WSDL-pohjaisista palveluista käyttää vanhempaa kuvan 6 mukaista 1.1 versioita. Vanhempi versio 1.1 on käytännössä käytettävissä ainoastaan SOAP:lla kun taas uudempi tarjoaa

mahdollisuuden käyttää myös kaikkia HTTP menetelmiä joten myös REST palvelut ovat mahdollisia. WSDL 2.0 käyttö on vielä vähäistä sille tehtyjen automatisoitujen työkalujen puutteen vuoksi.

Kuva 6. WSDL 1.1 tiedosto.

5 Liiketoimintaprosesseihin liittyvät tekniikat

5.1 BPM

BPM (Business Process Management) on holistinen suuntaus jolla pyritään ohjaamaan yrityksen toimintaa. BPM koostuu liiketoimintaprosessien analyysista, suunnittelusta, toteutuksesta, säädöksistä ja liiketoimintaprosessien kehityksestä [14, s 2]. Liiketoimintaprosessit ovat joukko tehtäviä minkä avulla saavutetaan liiketoiminnan tarkoitus yrityksessä. BPM korostaa liiketoiminnan tehokkuutta pyrkimällä kehittämään sitä teknologian avulla. Sen tarkoituksena on kehittää liiketoimintaprosesseja jatkuvasti joten sitä voidaan kutsua prosessin optimointi prosessiksi. BPM:stä väitetään myös että se olisi perinteisiä hierarkkisia liiketoiminnan hallintasuuntauksia tehokkaampia ja muovautuvampia. [15, s. 11.]

Yritysten liiketoimintaprosessien hallinnalle on kehityksen mittaamiseen kehittynyt kypsyysmalli BPMM (Business Process Maturity Model) mikä karkeasti kuvaa yrityksen liiketoiminta prosessien kypsyyden tasoa. Se on OMG:n (Object Management Group) ylläpitämä standardi [16, s. 73.].

Liiketoimintaprosessien hallinnan kypsyystasot:

1. Alkutilanne. Liiketoimintaprosesseja ei ole määritelty tai ne ovat vain tiettyä tarkoitusta varten.
2. Hallinnoitu. Joitakin prosesseja on määritelty ja dokumentoitu.
3. Standardoitu. Kaikki prosessit on määritelty ja dokumentoitu. BPM työkaluja käytetään mallintamiseen ja analysointiin. BPM käytetään päämäärän saavuttamiseen.
4. Ennakoitava. Prosesseja mitataan, kontrolloidaan ja ne on automatisoitu BPM systeemillä.
5. Innovatiivinen. Prosessien jatkuva ennakoiva kehitys jolla saavutetaan liiketoiminnan tavoitteet.

BPM ei ole suuntautunut pelkästään teknilliseen puoleen vaan se kattaa myös henkilöstön. Tärkeimmät osiot BPM:ssa yrityksille ovat käytettävyys, mallinnus valmiudet, raporttien generointi, ohjelmistointegraatio ja standardien yhdenmukaisuus. [1, s. 48.]

5.2 BPMN

BPMN (Business Process Model & Notation) on liiketoimintaprosesseille kehitetty standardoitu graafinen kuvauskieli. Sen on kehittänyt OMG joka myös vastaa ylläpidosta ja kehityksestä. Sen tehtävä on kuvata liiketoimintaprosessin toiminta alusta loppuun ihmisille ymmärrettävässä muodossa.[1, s. 1.]

BPMN on kehitetty eritoten yritysten kaupallisen ja teknisen puolen yhteiseksi kieleksi jota molemmat osapuolet ymmärtävät. Näin kaupallisen puolen henkilö joka tuntee liiketoimintaprosessin voi kuvata sen notaatiolla jonka tekninen puoli sitten implementoi palveluksi. [1, s. 1.]

5.3 BPEL

BPEL (Business Process Execution Language) tai toiselta nimeltään WS-BPEL (Web Service Process Execution Language) on XML-pohjainen liiketoimintaprosessien toiminnan suorituskieli. BPEL standardia ylläpitää ja kehittää OASIS (Organization for the Advancement of Structured Information Standards). Se kuvaa prosessin tehtäviä eritoten prosessin ja ulkoisten verkkopalveluiden kommunikointia. Sitä käytetään verkkopalveluiden koostamiseen, orkestrointiin ja koordinointiin. [17, s. 38–39.]

Toisin kuin BPMN on BPEL ymmärrettävää myös prosessimoottoreille joiden tehtävänä on suorittaa kuvausta vastaavaa prosessia. Koska BPEL on yksi tärkeimmistä standardoiduista suorituskielistä, sen käännös on määritelty BPMN-standardissa. Vaikka BPMN tulisi pystyä muuttamaan BPEL muotoon, BPEL:ssa on rajoituksia mitkä voivat estää joidenkin BPMN kuvausten kääntämisen. [18, s. 66.]

5.4 Apache ODE

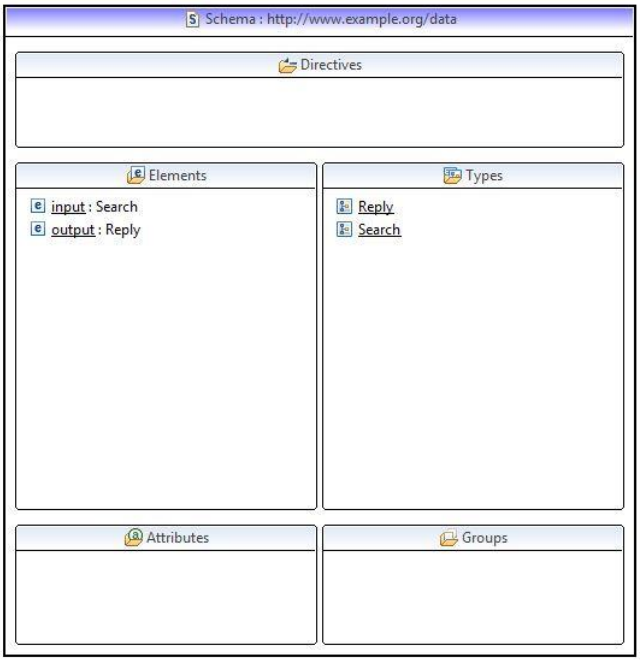
Apache ODE (Orchestration Director Engine) on ohjelmisto mikä suorittaa WS-BPEL-standardin mukaisesti laadittuja liiketoimintaprosesseja. Se kommunikoi verkkopalvelujen kanssa molempiin suuntiin sekä hallitsee datankäsittelyn ja virhetilanteet prosessin määrittelyn mukaan.

Apache ODE sisältää myös laajennuksia WS-BPEL spesifikaatiosta riippumatta. Näihin kuuluu muun muassa implisiittinen korrelaatio, aktiviteetin virheenhallinta, XPath, XQuery, ulkoiset muuttujat ja RESTful BPEL.

5.5 Intalio | bpms

Intalio | bpms on yksi käytetyimmistä vapaan lähdekoodin BPMS (Business Process Management Suite) ohjelmistoista. Siitä on saatavilla ilmainen yhteisöpohjainen ja maksullinen yrityspohjainen versio. Intalio | bpms koostuu kahdesta osasta palvelinpuolen ohjelmistosta Intalio | BPMS server ja ohjelmointiympäristöstä Intalio | Designer.

Intalio Designer on Eclipse-pohjainen ohjelmointiympäristö. Se on erityisesti laajennettu helpottamaan liiketoimintaprosessien mallinnusta sekä prosessien integroimista muihin sovelluksiin. Tämä on tehty mahdollisimman helpoksi sovelluksessa olevilla graafisilla työkaluilla. Designerilla voidaan helposti tehdä graafisia palveluita, prosesseja, tietokantayhteyksiä sekä teknisiä artefakteja esimerkiksi XML Schema (kuva 7), joista sovellus laati kuvan 8 mukaiset koodikieliset versiot.



Kuva 7. Graafinen XML-skeema.

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.example.org/data"
    xmlns:tns="http://www.example.org/data" elementFormDefault="qualified">

    <element name="input" type="tns:Search"></element>
    <element name="output" type="tns:Reply"></element>

    <complexType name="Search">
        <sequence>
            <element name="term" type="string"></element>
        </sequence>
    </complexType>

    <complexType name="Reply">
        <sequence>
            <element name="searchTerm" type="string"></element>
            <element name="message" type="string"></element>
        </sequence>
    </complexType>
</schema>
```

Kuva 8. Ohjelmallisesti luotu XML-tiedosto.

Designerillä mallinnettu BPMN-notaation mukainen liiketoimintaprosessi käännetään ensin BPEL:ksi mikä sitten julkaistaan Intalio | bpms serverille.

Intalio | BPMS server on Apache Tomcat pohjainen web-palvelin. Web-palvelimen lisäksi siihen kuuluu orkestrointimoottori Apache ODE sekä siihen on tehty muutoksia tukemaan liiketoimintaprosessien hallintaa.

6 Toteutus

6.1 BPMN ja User Task

Ensimmäinen tehtävä oli tutustua liiketoimintaprosessin mallinnuskielen notaatioon [1]. Dokumentti selvittää standardin ja kuinka liiketoimintaprosessi kuvataan.

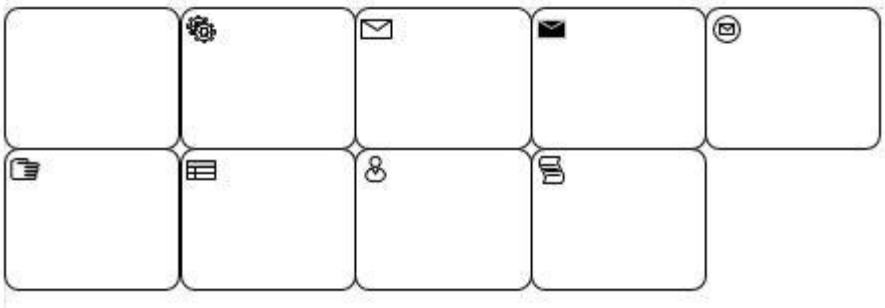
Dokumentista selviää että prosessin kuvaus on eräänlainen vuokaavio joka koostuu erilaisista tehtävistä. BPMN:n laajuuden vuoksi keskityin vain rajattuun osaan dokumentista tehtäviin (Task).

BPMN:ssa aktiviteettejä on erilaisia eri tilanteisiin mutta kaikilla on samanlainen graafinen pohja kuvassa 9. Tämä kuvaa yleistä aktiviteettiä mitä ei ole määritelty tarkemmin.

Kuva 9. BPMN yleinen aktiviteetti.

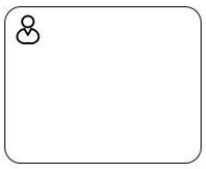
Tehtävät (Task) ovat aktiviteettejä joita ei ole purettu tai ei voida purkaa tarkemmin mallinnettuihin aliprosesseihin. Tehtäville on myös omat tarkemmin määritellyt kuvionsa joita on kahta erilaista. Käyttäytymistä kuvaavat kuvassa 10 ja tyyppiä kuvaavat kuvassa 11.

Kuva 10. BPMN käyttäytymistehtävät.



Kuva 11. BPMN tehtävätyypit.

Itse käyttäjäntehtävä (User Task) ei eroa yleisestä tehtävästä muutoin kuin että se osoittaa milloin prosessi tarvitsee ihmiskäyttäjän toimintaa prosessin edistämiseen. Käyttäjätehtävälle on notaatiossa oma merkintänsä kuvassa 12.



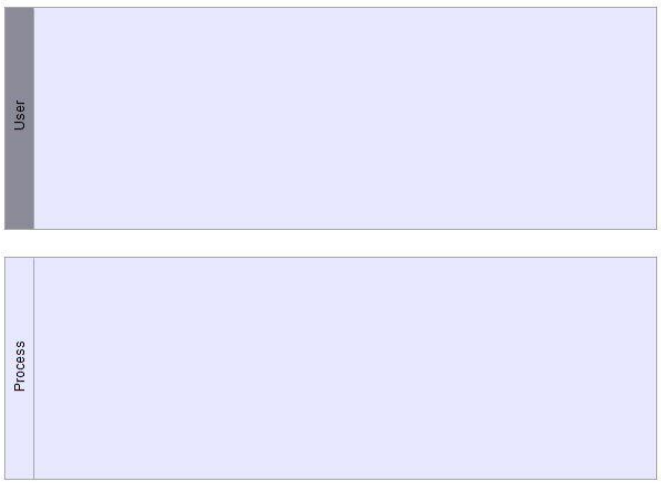
Kuva 12. Käyttäjätehtävä.

6.2 Liiketoimintaprosessin implementointi

6.2.1 Liiketoimintaprosessin kuvaus ohjelmointiympäristöllä

Liiketoimintaprosesseihin tutustumisen jälkeen lähdettiin tutkimaan mitä tapoja Intalio Designerissa on luoda prosesseja ja niihin liitettyjä tehtäviä.

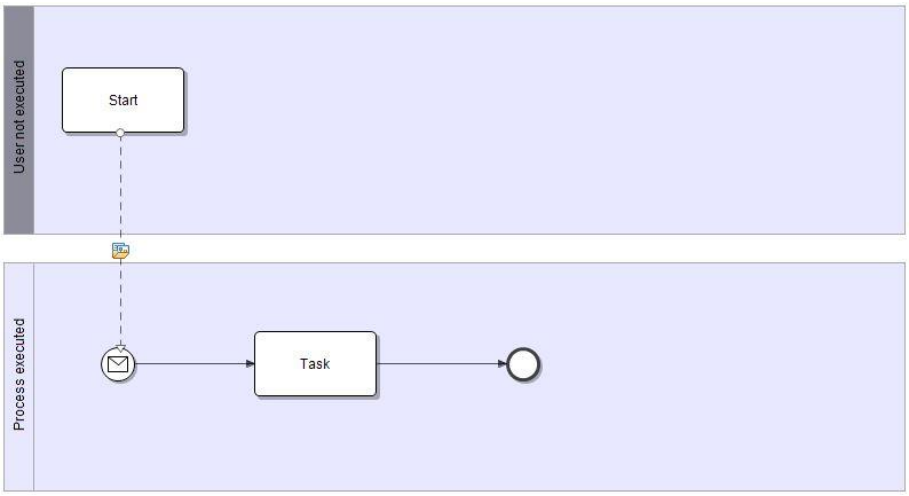
Ensimmäiseksi tutkittiin ohjelmistokehitysympäristön omaa tapaa tehdä käyttäjän antaman syötteen mukainen prosessi. Tässä päädyttiin tekemään mahdollisimman yksinkertainen prosessi mikä pelkästään ottaa vastaan käyttäjän antaman syötteen mutta ei tee sillä mitään. Prosessi koostuu kuvan 13 mukaisesta suoritettavasta ja suorittamattomasta osuudesta.



Kuva 13. Prosessin suoritusosat.

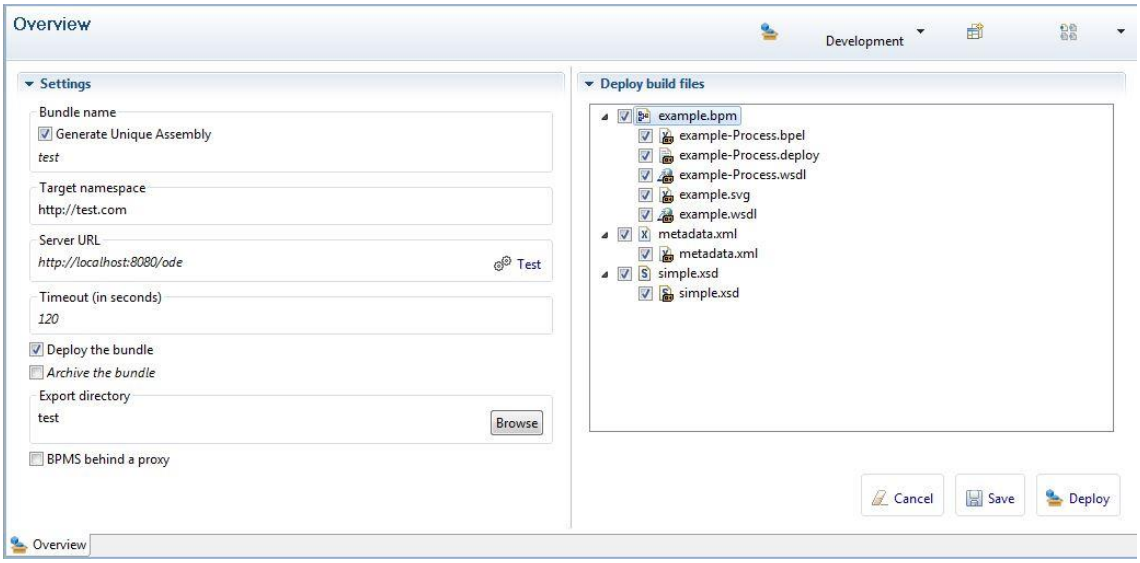
Prosessissa suorittamatonta osuutta käytetään palveluiden, syötteiden ja tietokanta yhteyksien liityntöjen kuvaamiseen. Tälle pohjalle rakentuu kuvan 14 mukainen prosessi

missä käyttäjä antaa yksinkertaisen tekstisyötteen. Tämä käynnistää prosessin, mikä ei sittemmin tee mitään, mutta kuvaa syötteen antamista prosessimoottorille. Toisin kuin BPMN-standardissa, missä käyttäjän tehtävät pitäisi merkitä kuvan 12 mukaan, ei Designerissä tämänlaista tehtävän merkintää löydy. Tämä todennäköisesti johtuu siitä, että prosessin ulkopuolisista syötteistä toteutetaan SOAP-pohjainen verkkopalvelu. Palvelun kuvaus löytyy palvelimella ODE-prosessimoottorin julkaistujen prosessien verkkopalvelujen osoitteesta http://localhost:8080/ode/deployment/services/ WSDL-tiedostona.



Kuva 14. Yksinkertainen prosessi käyttäjän antamasta syötteestä.

Prosessin julkaisu tapahtuu Intalio | bpms palvelimelle helposti Designerin julkaisutyökaluilla kuvassa 15.

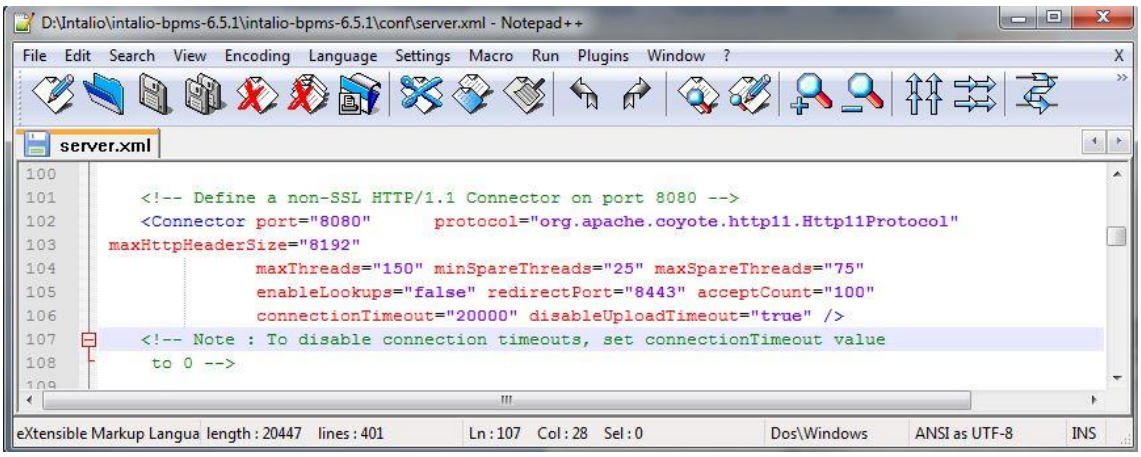


Kuva 15. Designer julkaisu ikkuna.

Designer tekee tässä tarvittavat käännökset ja luo tarvittavat tiedostot notaation pohjalta mikä sitten lähetetään prosessimoottorille mikä sijaitsee Server URL kohdassa.

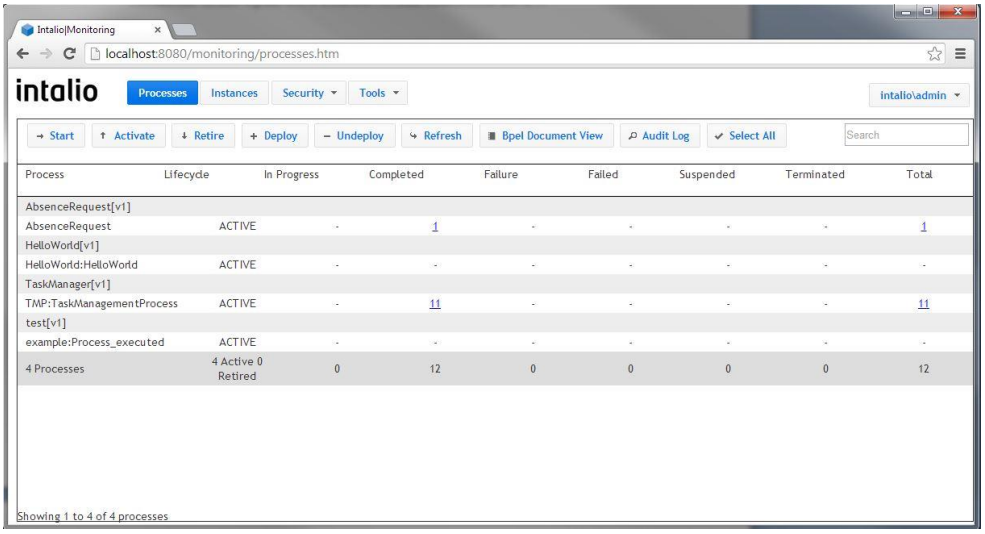
6.2.2 Liiketoimintaprosessin julkaisu

Intalio | BPMS palvelin sijaitsee asennuksen jälkeen oletuksena osoitteessa http://localhost:8080, joten mikäli palvelimella on useampia web-palvelimia, mitkä saattavat käyttää samaa osoitetta, täytyy se muistaa vaihtaa. Tämä konfiguraatio sijaitsee oletuksena intalio-bpms-6.5.1\conf\server.xml tiedostossa kuvassa 16.

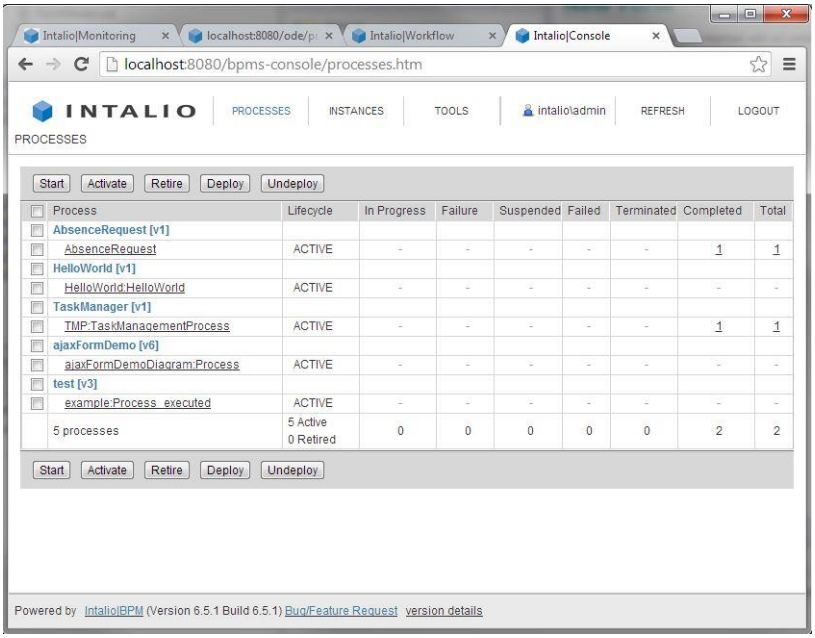


Kuva 16. Server.xml konfiguraatio.

Intalio | bpms -palvelimelle yhteyden saa verkkoselaimella aiemmin määriteltyyn osoitteeseen, tässä tapauksessa http://localhost:8080. Osoitteeseen yhdistämisen jälkeen verkkoselaimen pitäisi näyttää sisäänkirjautumisikkuna. Palvelimelle kirjauduttua aukeaa pääikkuna, mistä voi valita Intalion oman prosessieninstanssienhallinnan- tai prosessienhallinnanikkunat. Tällä hetkellä meitä kiinnostaa prosessinhallintaikkuna kuvassa 17. Hallintaikkuna näyttää kaikki prosessimoottorille julkaistut prosessit ja tietoja niistä. Prosesseille on myös hallintakonsoli kuvassa 18, löytyy palvelimelta osoitteesta http://localhost:8080/bpms-console, missä löytyy hieman enempi toimintoja prosesseihin liittyen. Ikkunassa näkyy myös aiemmin julkaistu yksinkertainen käyttäjän syötteen vastaan ottava prosessi test.

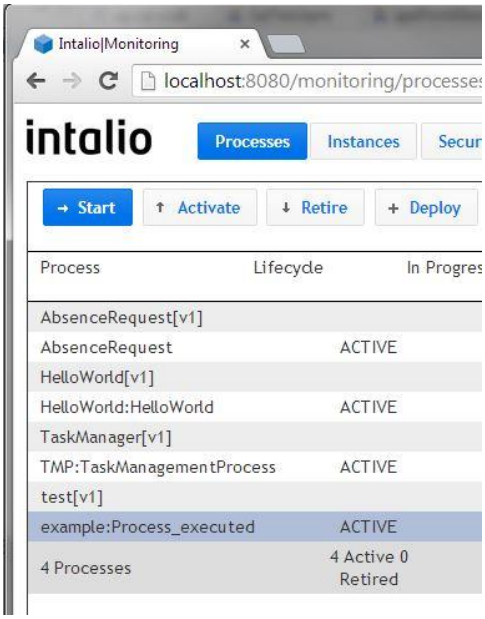


Kuva 17. Prosessien hallintaikkuna.

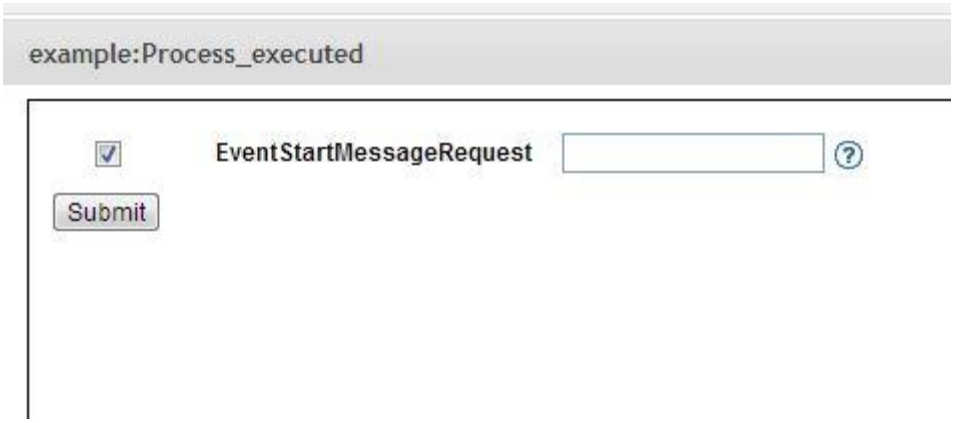


Kuva 18. Prosessien hallintakonsoli.

Julkaistua prosessia voidaan nyt testata ilman erillisiä ohjelmia tämän hallintaohjelman kautta. Valitsemalla haluttu prosessi ja käynnistämällä sen hallintaohjelmisto, kuvassa 19, tekee karkean, kuvan 20 mallisen, syötteenlukuikkunan, minkä jälkeen prosessista tehdään yksittäinen instanssi.

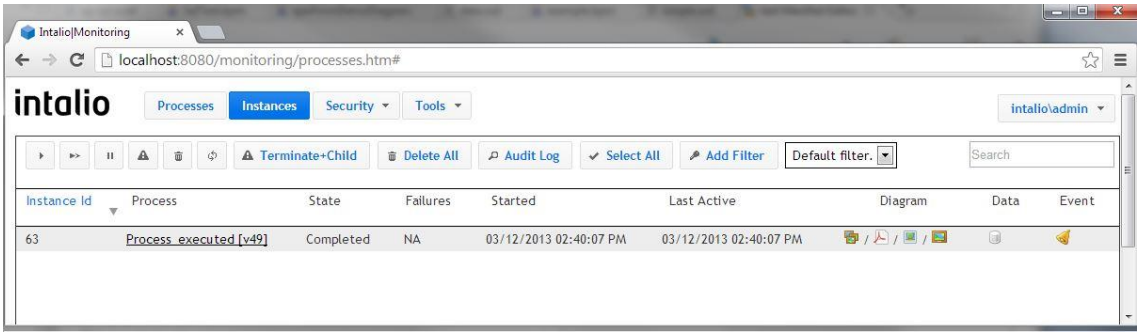


Kuva 19. Prosessin käynnistys.



Kuva 20. Prosessin syöteikkuna.

Koska prosessi oli hyvin yksinkertainen, sen instanssi valmistui heti syötteen saatuaan. Instanssi on prosessin yksi ilmentymä ja niitä voi olla samasta prosessista useita käynnissä yhtä aikaa. Näitä instansseja voidaan seurata instanssin hallinta ikkunasta kuvassa 21.



Kuva 21. Instanssin hallinta.

Instanssin hallintaikkunassa voidaan yksittäisen instanssin tilaa seurata myös notaation mukaisessa diagrammissa. Tässä yksinkertaisessa prosessissa tätä ei tarvita koska prosessin instanssi suoriutuu heti sen luomisen jälkeen mutta monimutkaisimmissa prosesseissa tästä on hyötyä mikä näkyy myöhemmissä prosesseissa. Myös instanssin dataa voidaan tarkastella data kohdassa mikä aukaisee oman ikkunan instanssissa olevasta datasta joita voi tarkastella lähemmin kuvassa 22.

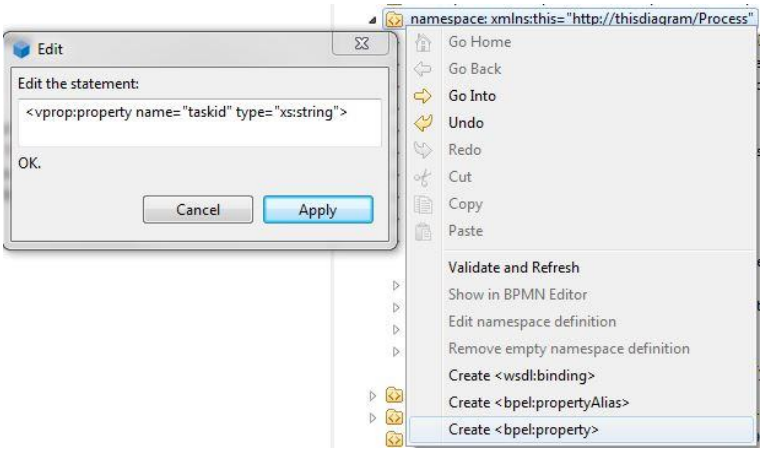
Kuva 22. Instanssin aloitusviestin data.

6.3 Laajennettu prosessi

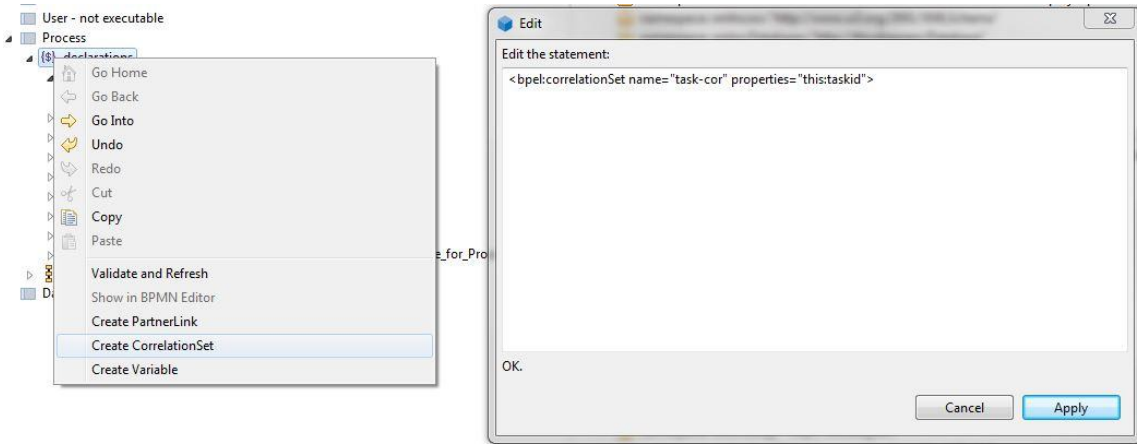
Jotta prosessia voitaisiin käyttää monimutkaisempiin tapauksiin, täytyy sitä laajentaa tästä yksinkertaisesta tapauksesta. Prosessin laajentamiseen tarvitaan paria tekniikkaa, korrelaatiota ja tiedon siirtämistä syötteen antavalle ohjelmistolle.

Prosessien instansseja emme voi palveluliitoksissa erotella toisistaan joten siihen tarvitaan jotain millä voimme kutsua prosessin tiettyä instanssia. Tähän löytyy prosessimoottorissa korrelaatio millä voidaan prosessiin määritellä instanssin yksilöivä tunnus.

Yhteisöversiossa ei löydy graafista työkalua tähän vaan se täytyy määritellä dataeditorilla. Tämä tapahtuu kolmessa osassa, korrelaatio ryhmästä, muuttujien kohdistaminen ryhmään ja BPEL-attribuutin määrittämisestä korrelaatiolle. Aluksi määritellään prosessimoottorille dataeditorilla, prosessin suoritettavalle osuudelle, BPEL-attribuutti mikä yksilöi prosessin instanssin kuvassa 23. Tälle täytyy määritellä prosessiin korrelaatioryhmä kuvassa 24.

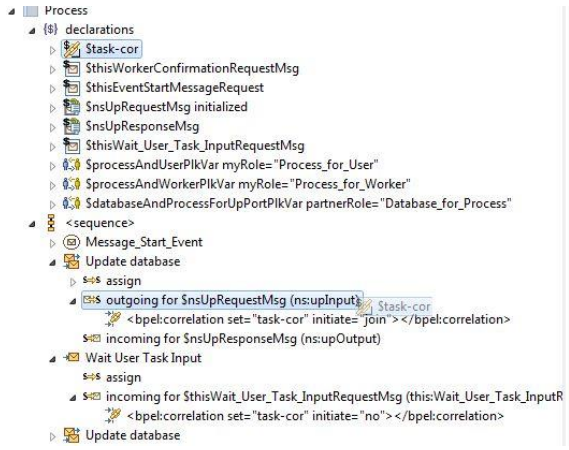


Kuva 23. BPEL-attribuutin määritys.

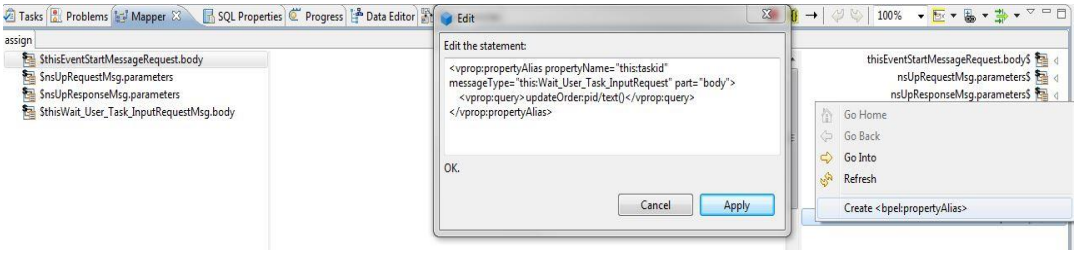


Kuva 24. Korrelaation ryhmän määritys.

Korrelaatio täytyy vielä assosioida haluttujen tehtävien kanssa kuvassa 25. Tehtäville täytyy vielä tehdä korrelaatiolle muuttujien vastaavuudet. Vastaavuuksien teko onnistuu helpoiten Mapper työkalun avulla missä prosessin dataa voidaan manipuloida graafisesti kuvassa 26.

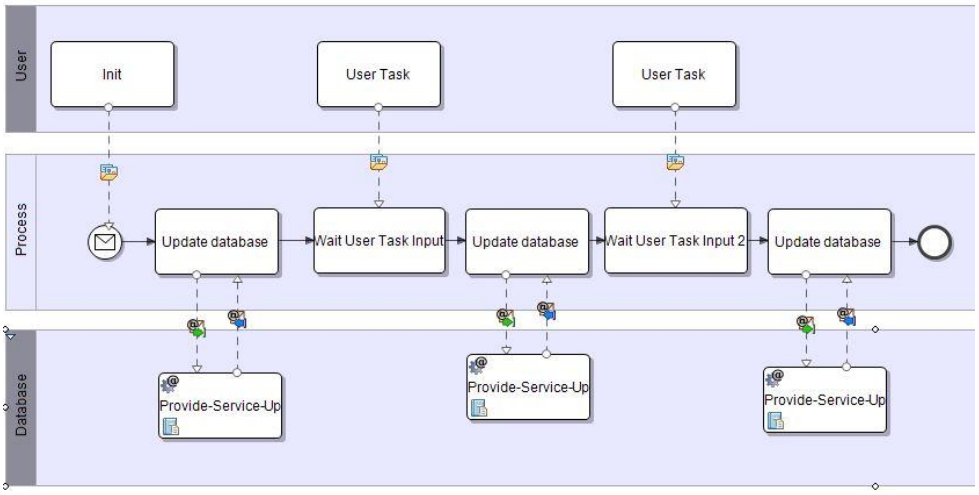


Kuva 25. Korrelaation assosioiminen tehtäviin.



Kuva 26. Korrelaatio vastaavuudet tehtävissä.

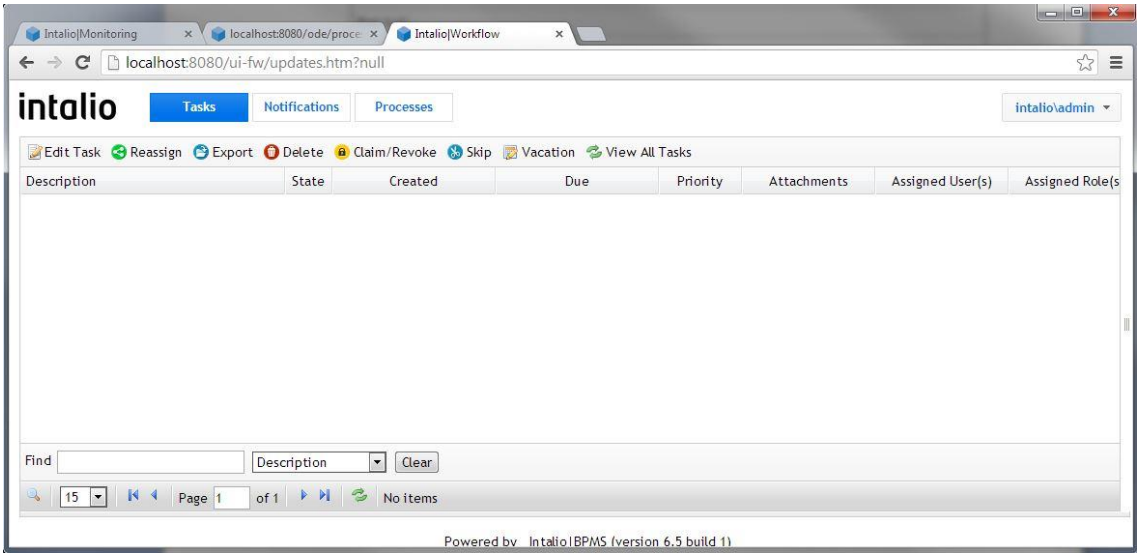
Näiden avulla saamme aikaan useita eri mahdollisia versioita miten dataa voidaan prosessista siirtää ulkoiselle käyttöliittymä järjestelmälle. Kuva 27 näyttää yhden mahdollisen tavan missä tiedon lähetys ulkoiselle ohjelmalle tapahtuu tietokannan välityksellä ja prosessille lähetys verkkopalveluiden kautta. Data voitaisiin myös lähettää ulkoiselle ohjelmistolle verkkopalvelunvälityksellä tai paluuviestinä prosessin aloitusviestiin.



Kuva 27. Esimerkki tietokannan välityksellä tapahtuvasta tiedonsiirrosta.

6.4 Intalio | Ajax

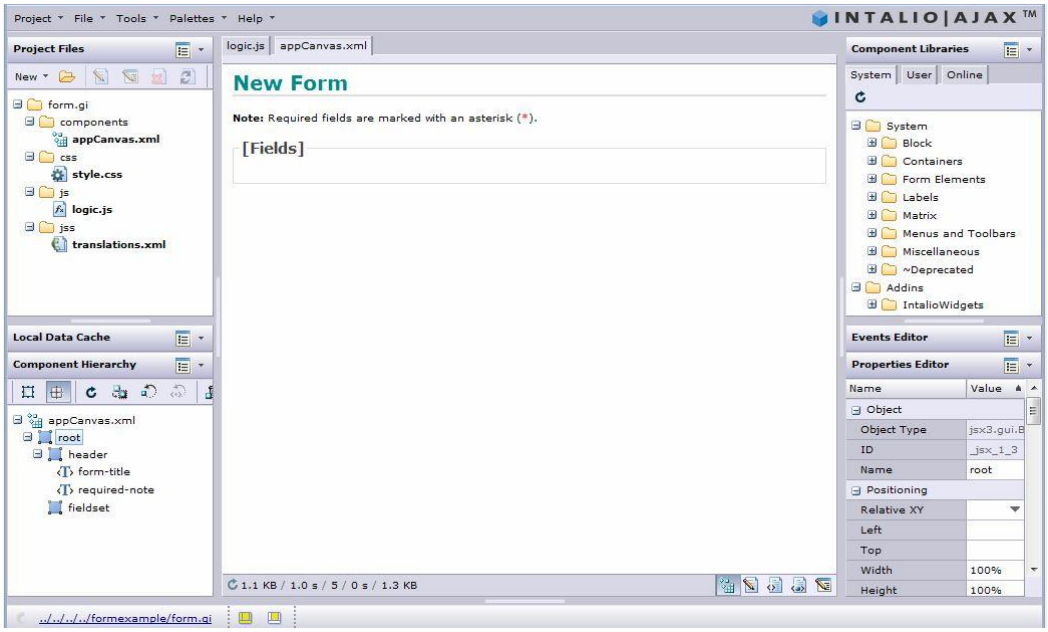
Seuraavaksi aloin tutkia, miten tehtävät toteutetaan Intalio | bpms ohjelmistossa löytyvillä ratkaisuilla. Tähän tarkoitukseen on palvelimella oma ohjelmistonsa ja prosessien seuranta prosessi. Se kulkee nimellä Task Manager tehtävienhallinta. Tehtävien hallintaan on oma sovelluksensa kuvassa 28, mikä löytyy palvelimelta osoitteesta http://localhost:8080/ui-fw.



Kuva 28. Tehtävienhallinta.

Tehtävienhallintaprosessia (TaskManagementProcess) voi tarkastella hallintakonsolin kautta mutta sen monimutkaisuuden ja koon takia emme perehdy tarkemmin siihen tässä tutkimuksessa. Tehtävienhallinta prosessin tehtävänä on hallita Intalion kehittämää lomakejärjestelmää. Lomakkeisiin tutustumisen jälkeen sanoisin että sitä voitaisiin muokata sisältämään myös muunkinlaiset prosessit.

Lomakepohjainen prosessi itsessään luodaan samalla tavalla kuin muutkin prosessit. Erona muihin prosesseihin siinä ilmestyviä tehtäviä ei piirretä suoraan notaatiolla vaan ne luodaan lomakkeista. Lomakkeiden luomiseen löytyy Designeristä oma editorinsa Intalio | Ajax kuva 29.

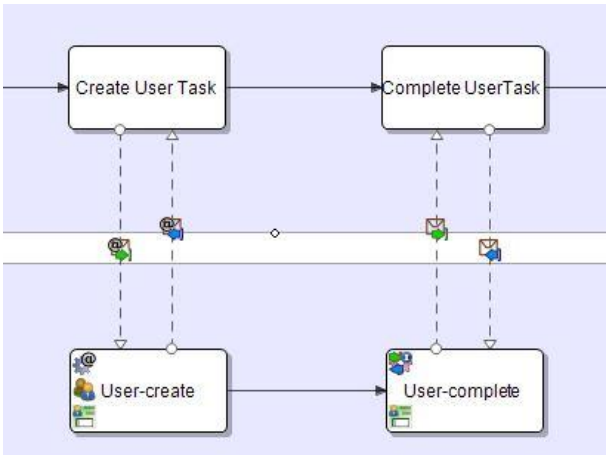


Kuva 29. Intalio | Ajax lomake editori.

Lomakkeet itse ovat XML-pohjaisia ja niihin voidaan kirjoittaa omaa logiikkaa JavaScriptillä. Lomakkeiden avulla voidaan luoda nopeasti tehtäviä ihmisille. Aluksi täytyy luoda editorilla lomake jossa on jonkinlainen syöte elementti kuten tekstikenttä. Lomakkeen luomisen jälkeen voidaan siitä luoda notaatioon tehtävä vetämällä lomake notaation päälle kuvassa 30. Tehtävän luomisen jälkeen se pitää vielä tehtävän asetuksista liittää käyttäjään tai ryhmään.

Kuva 30. Käyttäjätehtävän luonti lomakkeella.

Näistä meitä kiinnostaa kaksi ensimmäistä kohtaa PIPA (People Initiating Process Activity) ja PA (People Activity). PIPA-tehtävä voi olla yksinkertainen prosessin aloittava lomakkeen mukaisen syötteen antaminen prosessille tai samanlainen kuin PA mutta samalla prosessin aloittava. PA on kaksisuuntainen syötteen antaminen (kuva 31). Siinä prosessissa voidaan määritellä mitä tietoja käyttäjälle näytetään lomakkeella toisin kuin PIPA tehtävässä missä lomake on tyhjä. PA koostuu kahdesta tehtävästä kuvauksessa lomakkeen luomisesta ja lähetyksestä. Ensimmäisessä tehtävässä luodaan lomake ja esitäytetään halutut kohdat. Toisessa tehtävässä odotetaan käyttäjän syötettä minkä jälkeen lomake tuodaan takaisin prosessiin.

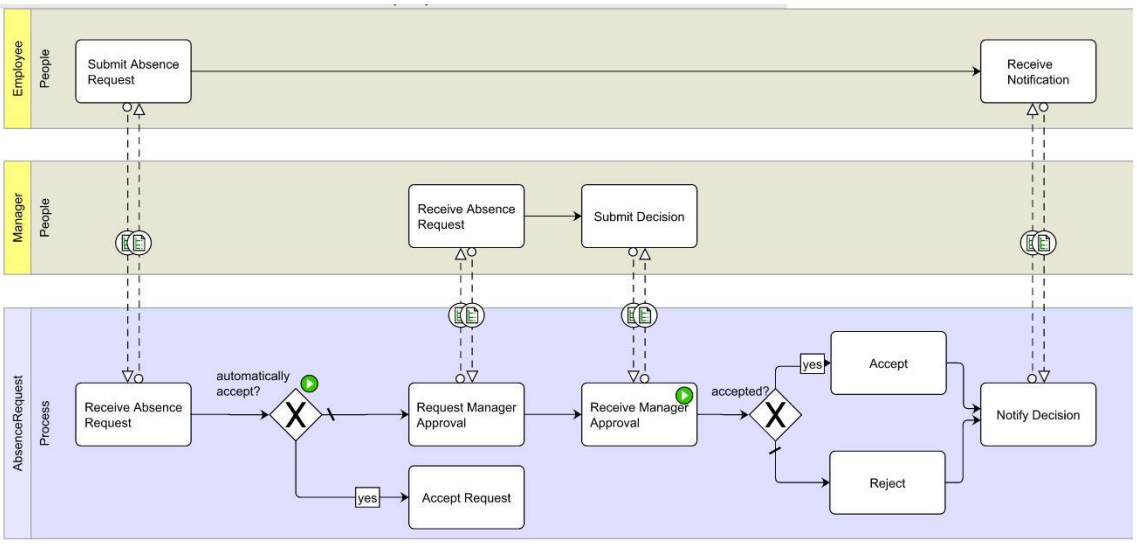


Kuva 31. PA tehtävä.

Lomakkeiden toimintaa on helppo kokeilla palvelimen mukana tulevan esimerkkiprosessin avulla, mikä simuloi poissaoloilmoitusta. Tästä lomakepohjaisesta prosessista luodaan kaksi ilmentymää instanssia yksi itse prosessista ja toinen hallintaprosessi kuvassa 32.

Kuva 32. Lomake prosessit.

Prosessin tilaa voidaan tarkastella notaation mukaisessa muodossa kuvassa 33. Vihreä nuoli osoittaa missä prosessin kohdassa prosessi on ja mikäli siihen liittyy jokin tapahtuma mitä se odottaa. Tässä tapauksessa prosessi odottaa poissaoloilmoituksen hyväksymistä jonka jälkeen se pääsisi etenemään.



Kuva 33. Intalio | bpms esimerkkiprosessin tila.

7 Tulokset

Tutkimuksessa käytettyyn ohjelmistoon Intalio | bpms tutustuttaessa saatiin selville että se on kehitetty luomaan ja hallitsemaan liiketoimintaprosesseja. Web-palvelin, minkä päälle Intalio | bpms palvelin on rakentunut, ei sovellu oikein muiden sovellusten kuin liiketoimintaprosessien ajamiseen. Prosessit suoritetaan palvelimen Apache ODE BPEL-prosessimoottorissa ja niihin viestitetään verkkopalveluilla. Kaikki prosessiin ulkopuolelta tulevat tiedot tulee syöttää verkkopalveluiden avulla tai tietokannan ja jonkinlaisen tapahtuman avulla, esimerkiksi prosessissa pyörivä ajastin, mikä tarkistaa tietokantaa säännöllisesti muutosten varalta.

BPMN on standardi millä voidaan graafisesti kuvata liiketoimintaprosesseja. Sen merkittävin tehtävä on poistaa markkinointipuolen ja teknisen henkilöstön välisiä kommunikointi ongelmia.

Käyttäjätehtävä on BPMN-notaatiossa erikseen määritetty tehtävä jonka tulisi prosessissa suorittaa käyttäjä. Intalio Designerilla tehdyssä prosessikuvauksessa ei käyttäjätehtäviä mallinneta aivan 2.0 version mukaisesti vaan yleistehtävinä. Tähän erona ovat Intalion omalla implementaatiolla tuotetut lomakepohjaiset prosessit. Näissä prosesseissa käyttäjätehtävät täytyy kohdistaa käyttäjään tai käyttäjäryhmään ja ne merkitään notaation mukaisella ihmistä muistuttavalla merkillä.

Projektissa oli alun perin tarkoitus myös laatia mahdollinen ulkopuolinen käyttöliittymä prosessin syötteitä varten mutta sen implementointi päätettiin jättää pois. Tämä johtui siitä että, riippumatta millä tekniikalla tahansa käyttöliittymä olisi tehty, joutuisivat ne kommunikoimaan BPEL-prosesssimoottorin kanssa verkkopalveluilla. Lisäksi Intalio | bpms sisälsi tarvittavat työkalut prosessien testaamiseen ilman ulkopuolisia ohjelmia.

8 Pohdinta

Työ onnistui muuten hyvin tosin aikataulusta todella paljon venyen. Tähän vaikuttivat ulkopuoliset seikat sekä ohjelmistojen epävakaisuus käytetyllä käyttöjärjestelmällä.

Vaikka työssä pääasiassa käytetty ohjelmisto Intalio | bmps oli päivittynyt vuoden alussa, ei se tuntunut olevan vakaa. Tämä todennäköisesti johtui päivityksistä ohjelmiston eri osiin, minkä takia ne eivät enää olleet täysin yhteensopivia vanhan version kanssa. Tämä aiheutti sen että ohjelmisto ei ollut vakaa ja työssä esitetyt prosessit eivät tutkijan mielestä ole täysin luotettavia.

Mikäli tutkimuksessa olisi käytetty maksullisia versioita ohjelmistoista, olisi niihin sisältynyt ohjelmistojen kehittäjän puolelta tukea. Tämä taas olisi vaikuttanut tutkimustulosten luotettavuuteen. Näin ollen tutkimus olisi syytä suorittaa uudestaan vakaammilla ohjelmistoilla.

Koska tutkimus rajattiin tiettyyn aiheeseen ja yhteen tuotteeseen eivät työssä käsittelyssä olleet tapaukset välttämättä toimi muille ohjelmistoille siirrettäessä. Tämä on ristiriidassa BPMN-notaation kanssa, sillä sen mukaan kuvauksia täytyisi pystyä käyttämään ohjelmistosta riippumatta.

Tutkimusta voisi lähteä kehittämään edelleen tutkimalla miten muissa ohjelmistoissa BPMN-notaation mukaiset prosessit toteutetaan ja vertailla niitä keskenään. Lisäksi voisi tutkia miten eri tekniikoilla käyttöliittymien verkkopalveluliitännät toteutetaan.

Lähteet

1. Object Management Group. Business Process Model and Notation (BPMN) Version 2.0. 2011. http://www.omg.org/spec/BPMN/2.0/PDF/. Luettu 1.12.2013
2. Harmon, Paul. Business Process Change : A Guide for Business Managers and BPM and Six Sigma Professionals (2nd Edition). Morgan Kaufmann. Burlington, MA, USA. 2007.
3. Juric, Matjaz & Pant, Kapil. Business Process Driven SOA using BPMN and BPEL. Packt Publishing Ltd. Olton Birmingham, GBR. 2008.
4. Porter, Michael E. Competitive Advantage: Creating and Sustaining Superior Performance. THE FREE PRESS, Simon & Schuster Inc. New York, NY 10020. 1985.
5. Taghizadegan, Salman. Essentials of Lean Six Sigma. Butterworth-Heinemann. Burlington, MA, USA. 2006.
6. Network Working Group. Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P. & Berners-Lee, T. Hypertext Transfer Protocol -- HTTP/1.1. 1999. http://www.w3.org/Protocols/rfc2616/rfc2616.txt. Luettu 1.12.2013.
7. Network Working Group, Berners-Lee, T., Fielding, R. & Masinter, L. Uniform Resource Identifiers (URI): Generic Syntax. 1998. http://www.ietf.org/rfc/rfc2396.txt. Luettu 1.12.2013.
8. W3C, Bray, T., Paoli, J., Sperberg-McQueen, C. M., Maler, E. & Yergeau, F. Extensible Markup Language (XML) 1.0 (Fifth Edition). W3C Recommendation. 2008. http://www.w3.org/TR/xml/. Luettu 1.12.2013.
9. Fawcett, Joe., Ayers, Danny. & Quin, Liam. R.E Beginning XML, 5th Edition (5th Edition). Wiley Somerset, NJ, USA. 2012.
10. W3C, Addison, P. Philips. Requirements for the Internationalization of Web Services. W3C Working Group Note. 2004. http://www.w3.org/TR/2004/NOTE-ws-i18n-req-20041116/. Luettu 1.12.2013.
11. Sandoval, Jose. RESTful Java Web Services: Master Core REST Concepts and Create RESTful Web Services in Java. Packt Publishing Ltd Olton Birmingham, GBR. 2009.
12. W3C, Gudgin, M., Hadley, M., Mendelsohn, N., Moreau, J., Frystyk Nielsen, H., Karmarkar, A. & Lafon, Y. SOAP Version 1.2 Part 1: Messaging Framework (Second Edition). W3C Recommendation. 2007. http://www.w3.org/TR/2007/REC-soap12-part1-20070427/. Luettu 1.12.2013.
13. Jayasinghe, Deepal & Azeez, Afkham. Apache Axis2 Web Services. Packt Publishing Ltd. Olton Birmingham, GBR. 2011.
14. Beckmann, Jason A. Business Issues, Competition and Entrepreneurship : Business Process Modeling : Software Engineering, Analysis and Applications. Nova Science Publishers, Inc. Hauppauge, NY, USA . 2011.
15. Patig, Susanne. BPM Software and Process Modelling Languages in Practice : Results from an empirical investigation. Frank & Timme GmbH. Berlin, DEU. 2011.
16. Object Management Group. Business Process Maturity Model (BPMM) Version 1.0. 2008. http://www.omg.org/spec/BPMM/1.0/PDF. Luettu 1.12.2013.
17. Juric, Matjaz B. & Krizevnik, Marcel. WS-BPEL 2.0 for SOA Composite Applications with Oracle SOA Suite 11g. Packt Publishing Ltd. Olton Birmingham, GBR. 2010.
18. Juric, Matjaz. & Pant, Kapil. Business Process Driven SOA using BPMN and BPEL. Packt Publishing Ltd. Olton Birmingham, GBR. 2008.

Lyhenteet

BPEL	Business Process Execution Language on prosessin kuvauksen suorituskieli.
BPMN	Business Process Model & Notation on graafinen kuvaus liiketoimintaprosessista.
BPMS	Business Process Management Solution tai Suite, on ohjelmisto liiketoimintaprosessien hallintaan.
HTTP	Hypertext Transfer Protocol on tiedonsiirto protokolla.
JSON	JavaScript Object Notation on kevyt tiedonsiirtoformaatti. Se on ihmisille helppoa ymmärtää ja koneille helppoa luoda ja lukea.
OASIS	Organization for the Advancement of Structured Information Standards on kansainvälinen konsortio, minkä tehtävänä on edistää verkkopalveluiden ja elektronisen kaupankäynnin standardeja
ODE	Orchestration Director Engine on BPEL kielisiä prosessikuvauksia suorittava moottori.
OMG	Object Management Group on konsortio, joka on perustettu oliopohjaisten hajautettujen järjestelmien standardien asettamiseen.
REST	Representational State Transfer on standardoitu HTTP-pohjainen tiedonsiirto menetelmä.
SOA	Service Oriented Architecture on arkkitehtuurityyli sovelluksille.
SOAP	Simple Object Access Protocol on tiedonsiirtotapa mikä käyttää XML-tiedostoja tiedonsiirtoon.
URI	Unified Resource Identifier resurssin yksilöivä osoitin.
W3C	World Wide Web Consortium on organisaatio mikä määrittelee ja hallinnoi WWW:n standardeja.
WSDL	Web Services Description File on XML-tiedosto joka kuvaa verkkopalvelun toiminnan.
XML	eXtensible Markup Language on standardoitu tiedostomuoto, jota käytetään tiedonsiirtämiseen.