

QPR-integrointiraportti

Raportissa kuvataan tarvittavat muutokset JY:n käyttämään QPR-prosessikuvauspohjaan, jotta prosessimallit voidaan siirtää ProcMinerin ja QPR:n välillä mahdollisimman pienillä muutoksilla. Lisäksi tarkennetaan ohjeita QPR-mallinnukseen, prosessien kuvauskäytäntöjä ja esitetään toteutusratkaisuja.

1 Johdanto

ProcMiner on IT-tiedekunnassa käytössä oleva prosessien hallintasovellus, jolla on vuosien 2005-2006 aikana mallinnettu yli 150 eritasoista prosessikuvausta. ProcMiner tukee prosessien html- ja pdf-julkaisua ja sisältää monipuoliset hakutoiminnot. Prosessit on kuvattu tekstimuodossa XML-pohjaisella kuvauskielellä. Lisäksi sovelluksessa on lukurajapinnat Microsoft Visioon, Microsoft Exceliin ja QPR:ään.

QPR on Jyväskylän yliopistossa käytössä oleva prosessimallinnussovellus, joka mahdollistaa prosessien visuaalisen mallintamisen ”uimaratakaavioina”. ProcMinerin tietomalli on QPR:n vastaavaa ilmaisuvoimaisempi, joten 2-suuntainen, tiedot säilyttävä muunnos edellyttää tarkasti määriteltyjä mallinnuskäytäntöjä. Tietojen siirrettävyys on tarpeellista, koska monet IT-tiedekunnassa mallinnetut prosessit sopivat sellaisenaan tai pienillä muutoksilla myös muiden tiedekuntien käyttöön. Toisaalta alun perin QPR:llä mallinnettujen prosessien siirto ProcMineriin parantaa niiden haettavuutta ja vertailtavuutta.

Prosessien mallinnus on vasta ensimmäinen askel prosessien hyödyntämisessä. Yhtenäiset ja ymmärrettävät prosessimallit tukevat yleistä laatuajattelun kehittämistä, mutta toimivat myös suoraan työohjeina prosessin toimijoille. ProcMinerin avulla prosesseista voidaan koostaa organisaatiokohtaisia raportteja monella eri tarkkuustasolla, mikä auttaa kasvavan prosessimäärän hahmottamisessa. Lisäksi yksittäisten dokumenttien ja roolien esiintymiä voidaan laskea prosessi- tai aktiviteettikohtaisesti, jolloin voidaan arvioida toimijoiden työ määrää tai prosessin automatisointitarvetta.

2 Ohjeita QPR-mallinnukseen

ProcMinerin tietomalli on QPR:n vastaavaa ilmaisuvoimaisempi, joten tietojen siirto edellyttää lisäyksiä JY:ssä vuonna 2006 käytössä olleeseen malliin. Uudet mallinnuselementit vaativat hieman aiempaa enemmän mallinnustyötä, mutta toisaalta prosessikuvausten haettavuus, ymmärrettävyys ja ylläpidettävyys paranevat. Uusien elementtien tarpeellisuutta on testattu käytännössä IT-tiedekunnassa vuosien 2005-2006 aikana, kun mallinnuskieltä on kehitetty rinnakkain prosesseja mallinnettaessa ja vertailtu muihin prosessimallinnusvälineisiin. Luvussa ohjeistetaan prosessien mallintamista huomioiden muokatun tietomallin uudet ominaisuudet. Muutokset on kuvattu tarkemmin luvussa 3.2.

Muokattu tietomalli, esimerkkejä prosessien kuvauskäytännöistä sekä tyhjä mallipohja ovat [www:ssä](http://www.jyu.fi/prosessit) html-sivustona ja suoraan QPR:n omassa formaatissa. Samassa hakemistossa on myös JY:n hallinto- ja tukipalveluiden prosessikuvaus alkuperäisessä ja ProcMinerin muotoon muunnatussa muodossa (*korjattu* viittaa prosessin 1.4.2.2. Palkka-aineiston sisäänluku yhden prosessiaskeleen kirjoitusasun muutokseen, joka ei näy loppukäyttäjälle, mutta on välttämätön ProcMiner-muunnoksessa – muuten prosessit ovat alkuperäisessä 29.12.2006 mallinnetussa muodossaan).

<http://prosessit.it.jyu.fi/qpr/>

Useimmille elementeille yhteisiä uusia metatietokenttiä (löytyvät mallinnuselementin ponnahdusvalikon *Properties*-dialogista) ovat seuraavat:

- Tekstikenttä URL, johon voi merkata kyseiseen prosessiin, prosessiaskeleeseen, dokumenttiin, organisaatioyksikköön tai tietojärjestelmään liittyvän www-osoitteen. Tätä voidaan käyttää esimerkiksi viittauksessa prosessissa toimivan organisaation web-sivulle tai dokumentin sähköiseen versioon.
- Monirivinen tekstikenttä *Description*, johon voidaan merkitä mallinnuselementistä yleisiä kommentteja ja tarkempaa kuvausta, joka ei näy suoraan html- ja pdf-julkaisuissa. *Description*-kentän käyttö on erillisiä *Note*-muistilappuja parempi tapa, koska kommentti voidaan kohdistaa erityisesti tiettyyn mallinnuselementtiin (prosessin yleistä kommentointia varten voidaan käyttää itse prosessielementin vastaavaa kenttää).

2.1 Prosessin yleiset tiedot

Prosessi kuvaa tietyn vakiintuneen toimintatavan yleiset ominaisuudet ja mahdollisesti oletusarvoisen työnkulun, prosessi-instanssi on tietyn yksikön toteutus prosessista. Prosessi-instanssit mahdollistavat monen eri organisaation prosessien sisällyttämisen samaan malliin niin, että mahdollisimman suuri osa yhteisistä tiedoista on yhdessä paikassa. Prosesseilla on seuraavat metatietokentät (löytyvät prosessisymbolin ponnahdusvalikon *Properties*-dialogista, omistajaa lukuunottamatta *Custom Attributes* -välilehdeltä): omistaja, asiakas, tarkoitus, esiehdot, jälkiehdot, syötteet, tuotokset, delegointitiedot, asiakasvaatimukset, menestystekijät, rajoitteet, kehitysehdotukset, ongelmat, URL. Prosessi-instansseilla on lisäksi toteuttajaorganisaatio. Prosessiin voidaan liittää myös siihen yleisesti liittyviä dokumentteja dialogin *Information items* -välilehdeltä.

Suurin osa ominaisuuskentistä on otettu käyttöön Kaari-projektista, mutta mukana on myös muutamia uusia. Tärkeimpiä kenttiä ovat *omistaja* (organisaatio tai rooli, joka vastaa prosessin läpiviennistä kokonaisuutena) ja *asiakas* (organisaatio tai rooli, jonka takia prosessia suoritetaan, usein sen käynnistäjä. Esim. useissa opintohallinnon prosesseissa *asiakas* on opiskelija). *Delegointitiedot*-kentällä voidaan kuvata vapaamuotoisesti, kenelle prosessin käytännön suoritus on delegoitu organisaatiossa. Tällöin prosessin omistajaksi voidaan merkitä mahdollisimman pysyvä rooli (esim. *laitoksen opintoasioista vastaava johtaja*, joka delegointitiedoissa on merkitty varajohtajan vastuulle). Kaikkien kenttien täyttäminen ei ole välttämätöntä, erityisesti jos kyseessä on lyhyt matalan tason aliprosessi. Perusteellinen merkkaukset parantaa kuitenkin prosessien haettavuutta ja kenttien tietoja voidaan käyttää suoraan prosessikäsikirjan luonnissa.

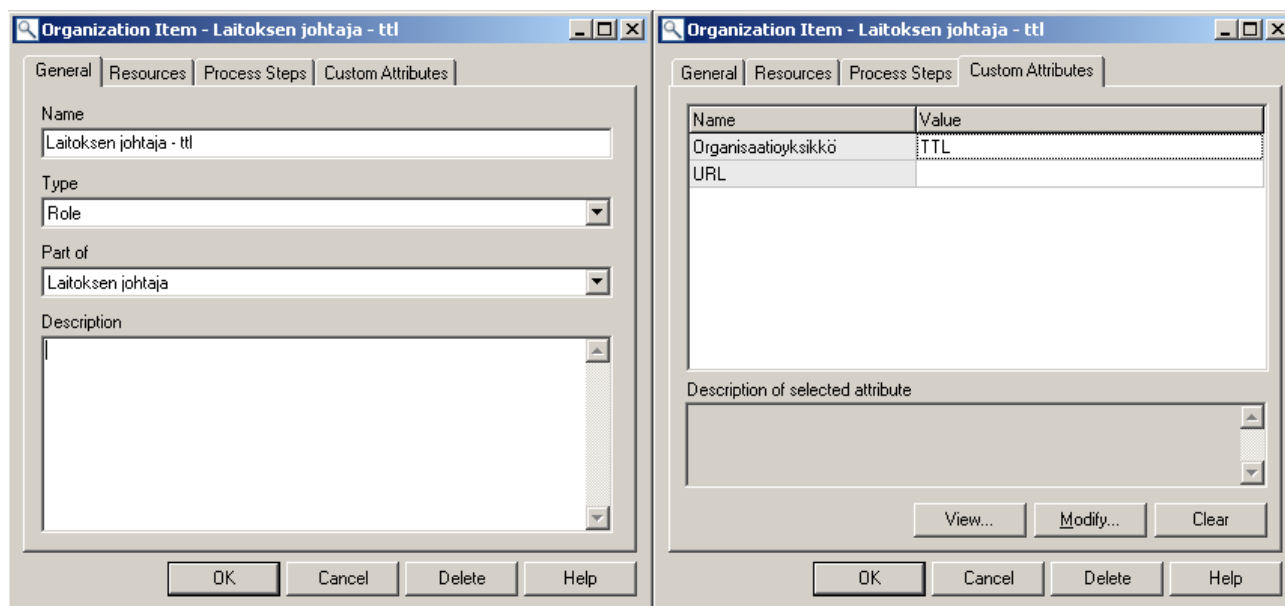
Metatietokenttien lisäksi prosessi sisältää työnkulun kuvauksen uimaratakaaviona, jossa itse työnkulku kuvataan aktiviteeteista, kontrolli- ja tietovirroista ja muista mallinnuselementeistä koostuvana rakenteena. Jos mahdollista, jokainen aktiviteetti sijaitsee omalla uimaradallaan, joka liittyy johonkin organisaatioon tai rooliin. Uimaradoilla kuvataan prosessin toimijat. Mallin eheys edellyttää, että jokainen kontrolli- ja tietovirta liittyy molemmista päistään johonkin mallinnuselementtiin.

2.2 Rooli- ja dokumenttihierarkiat

Prosessien uimaratojen toimijat, prosessien omistajat ja muut prosessimallien metatiedoissa mainitut organisaatioyksiköt sijoitetaan keskitetysti roolihierarkiaan, jolloin samaa roolia voidaan käyttää eri prosesseissa. Koska roolihierarkia on yhteinen kaikille prosesseille, niiden määrittelyssä on noudatettava erityistä tarkkuutta – varsinkin, jos samaa mallia muokataan monella eri laitoksella. Omia

rooleja ei pidä keksiä perusteettomasti, eikä samaa roolia tule kopioida moneen kertaan, jos sellainen on jo valmiina roolihierarkiassa.

Roolihierarkiaa tarkennetaan ottamalla käyttöön QPR:n tyyppikenttä (samassa *Properties*-dialogissa kuin prosesseilla), jonka avulla roolin voidaan määritellä olevan mm. organisaatioyksikkö tai virkanimike. Tämän avulla prosessikaavioiden uimaratoja voidaan tarkastella raportteja generoidessa joko yleisellä organisaatiotasolla tai tarkennetulla roolitasolla. Lisäksi virkanimikkeille tai muille yksittäistä henkilöä tai tiimiä vastaaville rooleille merkitään uutena metatietokenttänä *organisaatioyksikkö*. Tällöin voidaan määritellä "laitoksen johtaja" eri organisaatioille, mutta pitää hakuja varten rooli samana. Tyyppi- ja organisaatioyksikkö-kenttiä on havainnollistettu kuvassa 1.

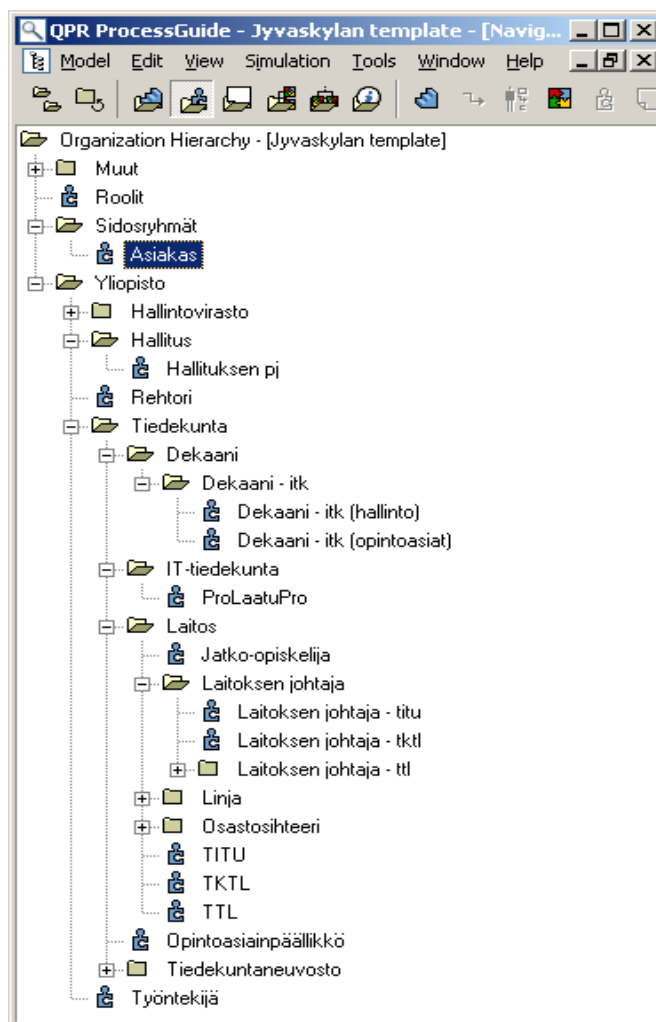


Kuva 1: Roolien uudet metatietokentät: tyyppi, organisaatioyksikkö ja URL.

Roolien hierarkkisuuuutta hyödynnetään niin, että tietyn organisaation aliorganisaatio (tai organisaation "instanssi" - esim. tietotekniikan laitos merkitään laitoksen alle) merkitään puussa alemmalle tasolle. Virkanimikkeet merkitään sen organisaation alle, johon ne ensisijaisesti liittyvät, mikä määrää myös roolin oletusarvoisen *organisaatioyksikön*. Jos roolista halutaan merkitä organisaatiokohtaisia ilmentymiä, ne merkitään suoraan alkuperäisen roolin alle ja muutetaan *organisaatioyksikkö*. Esimerkiksi "laitoksen johtaja - ttl" on "laitoksen johtaja"-roolin alla. Jos roolilla on vielä vastuualueen mukaan tarkennettuja tehtäviä, voidaan tehdä vielä toisen tason alirooli "laitoksen johtaja - ttl (opintoasiat)". Roolin ei tarvitse olla yksittäinen eikä edes tunnettu henkilö, mutta sen pitäisi yleensä voida edustaa eri organisaatioita (tai organisaation yksiköitä) - esimerkiksi jokaisessa tiedekunnassa on dekaani. Rooliin voi kuulua useampiakin henkilöitä (esim. tiedekuntaneuvosto, johtoryhmä), mutta prosessikuvauksessa rooli toimii yksikkönä. Kuva 2 on esimerkki tarkennetusta roolihierarkiasta.

Jos QPR:llä kopioidaan olemassa oleva prosessi uudeksi prosessiksi, ohjelma esittää seuraavan kysymyksen: *Selection contains stores/organization units. Press Yes to create copies of those elements. Press No to instantiate existing elements.* **Tähän on ehdottomasti vastattava "No"** - muuten ohjelma luo rooleista tarpeettomat kopiot, jotka sijoittuvat roolihierarkian huipulle. Saman roolin esiintyminen roolihierarkiassa moneen kertaan vaikeuttaa hakuja eikä ylimääräisiä rooleja pysty ko-

pioinnin jälkeen helposti korvattua alkuperäisiksi rooleiksi. Ongelma esiintyy JY:n hallinto- ja tuki-palveluiden prosessikuvauksissa useiden roolien kohdalla.

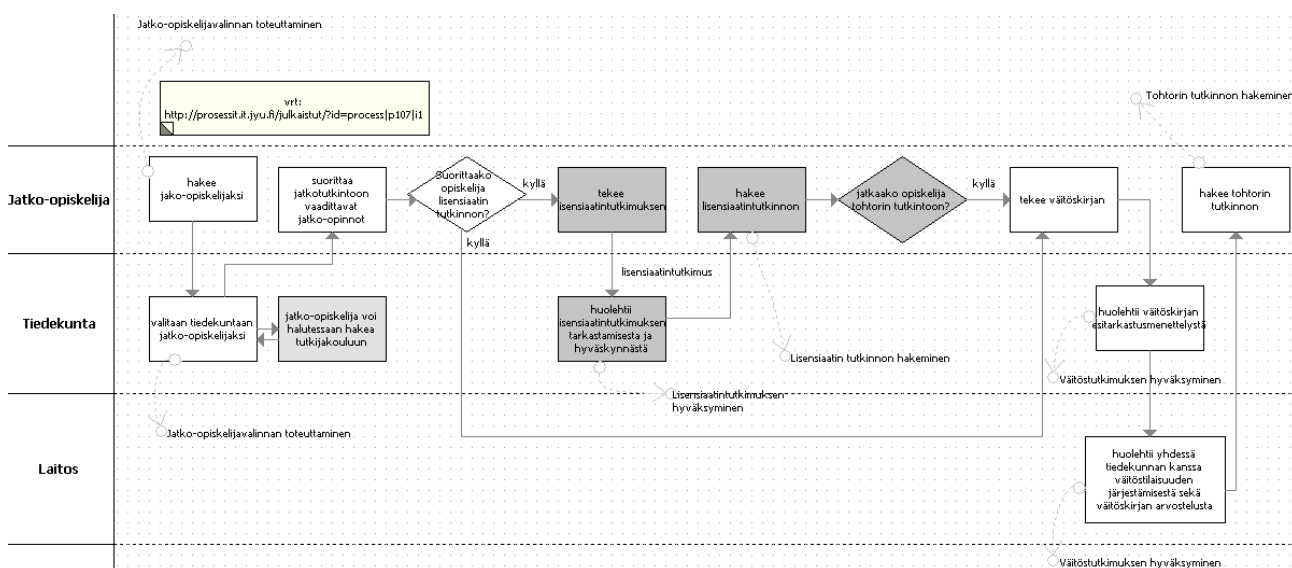


Kuva 2: Tarkennettu roolihierarkia.

Dokumentti- tai informaatioyksikköhierarkia on toinen roolihierarkiaa muistuttava rakenne, jonne merkitään organisaatiossa yleisesti käytössä olevat dokumentit, lomakkeet sekä tiettyihin prosesseihin vaikuttavat lait, johtosäännöt ja muut prosessin aikana esiintyvät eroteltavissa olevat informaation tyypit. Dokumentteja käytetään tietovirroissa tai prosessin yleisissä metatiedoissa. Roolihierarkian tapaan dokumenttihierarkia on kaikille yhteinen. Dokumenttien järjestämistä helpottaa jakamalla hierarkia teeman mukaisiin dokumenttiryhmisiin. Dokumenttiryhmää ilmentämään voidaan käyttää *Properties*-dialogin tyypikentää (esim. arvoa *Other document* -tyhjä tai muu arvo merkitsisi tavallista dokumenttia). IT-tiedekunnassa käytettyjä ylimmän tason dokumenttiryhmiä ovat esim. *opiskelijan dokumentit*, *henkilökunnan dokumentit* sekä *lait ja asetukset*, mutta täsmälliset dokumenttiryhmät tulee määritellä yhteisesti. Jos tietystä dokumentista halutaan myös organisaatiokohdaisia toteutuksia, dokumenteille voisi määritellä myös oman ”organisaatioyksikkö”-tyylisen metatietokentän roolien tapaan. Tälle ei kuitenkaan toistaiseksi ole ollut tarvetta.

2.3 Prosessin työnkulku

Yksittäisten prosessien työnkulun kuvauksen osalta mallinnusperiaate on pääpiirteissään sama kuin QPR:ssä oletuksena käytetty, mutta mukana on uusia mallinnuselementtejä. Merkittävimpänä uutena asiana on *abstraktiotasojen* käsite. Jokainen prosessin työnkulkuun liittyväle toiminto- ja päätös- askeleelle voidaan määrittää, onko kyseinen askel *ensisijainen* (taso 1), *tarkentava* (taso 2) vai *poikkeus* (taso 3). Eri tasot on värikoodattu ja niille on määritelty omat QPR-mallinnuselementtinsä. Esimerkkinä abstraktiotasojen avulla mallinnetusta prosessista on *Jatko-opiskelijan elinkaari* (kuva 3).



Kuva 3: Abstraktiotasot prosessissa.

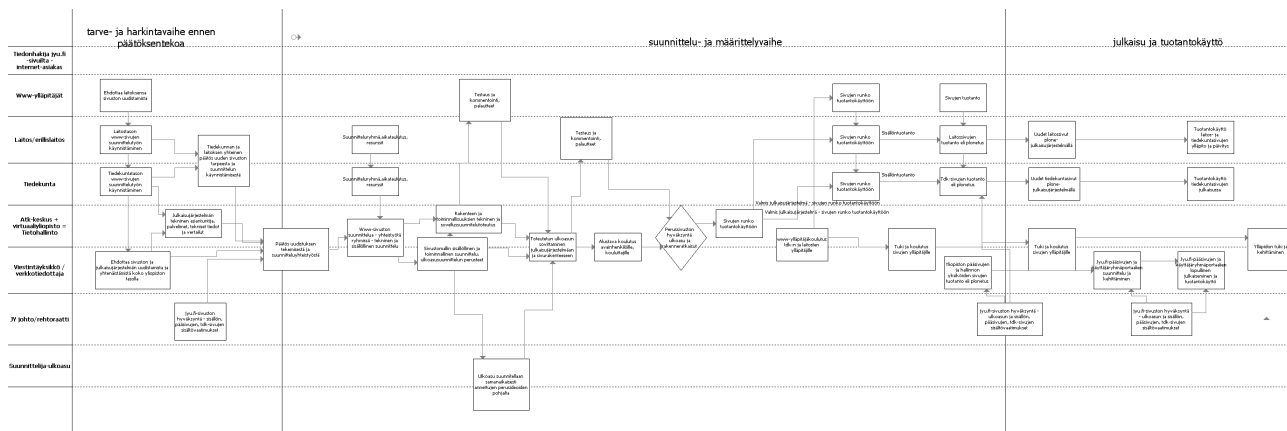
Tasolla 1 kuvataan prosessin oletusarvoinen ja ”luonnollisin” kulku, joka sisältää prosessin läpivien- nin kannalta keskeisimmät elementit. Hyvä käytäntö on mallintaa 1-tason elementit yhdeksi ketjuk- si, jolloin prosessin yleinen kulku selkiytyy. ProcMinerillä puolella 1-tason mallinnuselementeistä koostuvaa toimintoketjua kutsutaankin prosessin *Pääsekvenssiksi*. Koska useissa prosesseissa esiin- tyy tästä huolimatta vaihteita, joita ei voi helposti esittää yhtenä ketjuna, tällaiset *oleelliset* askeleet merkitään tason 2 ”tarkentavina” elementteinä. Tasolla 3 ovat puolestaan askeleet, jotka liittyvät sel- keästi poikkeustapauksiin, mutta joista prosessin toimijoiden on hyvä olla tietoinen.

Generoitaessa prosesseista laajaa yleiskatsausta riittää useimmiten käsitellä vain tasoja 1 ja 2. Varsi- naisten toimintaa kuvaavien askeleiden lisäksi useissa prosesseissa tarvitaan prosessiaskleita ku- vaamaan dokumentin tai muun tiedon siirtymistä toimijalta toiselle (tai esim. tietojärjestelmään), vaikka tiedon vastaanottaja ei toimisi muuten aktiivisesti prosessissa. Nämä ”tietovirran päät” mer- kitään tyhjällä toimintoaskeleella ja alemmalle abstraktiotasolle kuin tietovirran lähtöaskel.

Teknisistä syistä (katso luku 3.3) prosessin kaikilla tasoilla tulee pyrkiä ketjuihin prosessielemen- teissä. Toisin sanoen jokaisesta aktiviteetista tulisi samalla abstraktiotasolla olla korkeintaan yksi lähtevä (ja vastaavasti tuleva) kontrolli- tai tietovirta. Ketjujen avulla prosessin elementit voidaan numeroida ja järjestää yksikäsitteisesti ja kaavion lisäksi voidaan muodostaa tekstimuotoinen ku- vaus. Tämä saattaa olla kaaviomuotoa selkeämpi hahmottaa, jos prosessi on pystytty jakamaan mah- dollisimman pieneen määrään ketjuja.

Abstraktiotasojen huomiointi ei ole mallintajalle pakollista, mutta niiden käyttäminen on suositelta- vaa prosessin ymmärrettävyyttä ajatellen. Kuvaavana esimerkkinä tästä on viestinnän prosesseissa

oleva www-sivujen uudistusprosessi, joka todennäköisesti selkiytyisi, jos mallissa otettaisiin abstraktiotasot käyttöön. Prosessi on esitettyä kuvassa 4 QPR:ssä ja kuvassa 5 ProcMineriin tuotuna.



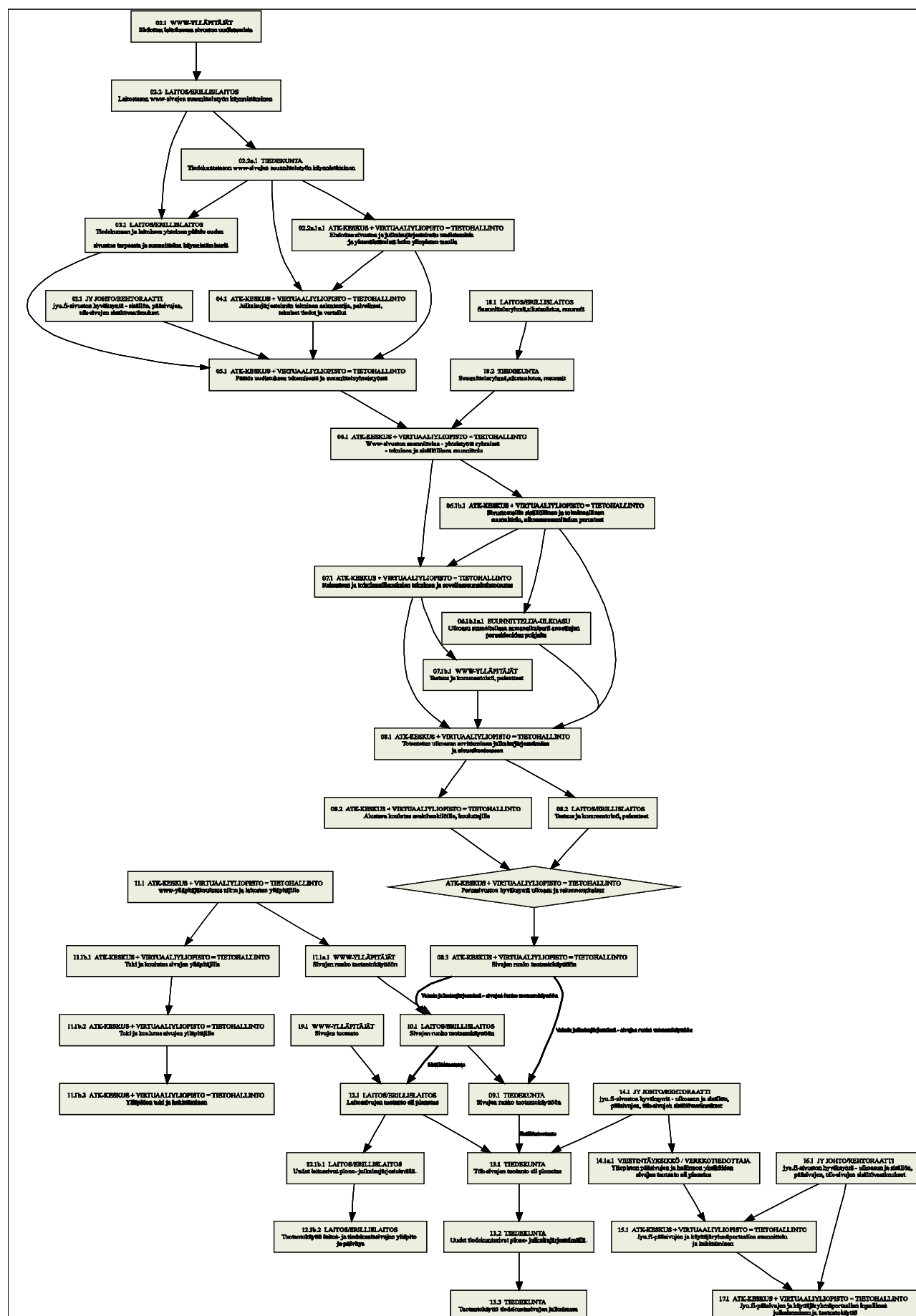
Kuva 4: WWW-sivujen uudistusprosessi (QPR).

Mallin eheys edellyttää abstraktiotasojen huomioonläänsi, että **jokaisen kontrolli- tai tietovirran on oltava kiinnittyneenä mallinnuselementtiin** virran molemmista päistä. Jos näin ei ole, ketjuja ei voida muodostaa luotettavasti ja prosessista generoitu kaavio esitetään väärin. Kiinnittymiseen on kiinnitettävä huomiota erityisesti silloin, kun kaaviossa siirretään useita elementtejä kerrallaan.

Jokaisella prosessilla tulisi olla yksi alku- ja päätepiste. Monien prosessien osalta nämä pisteet voidaan päätellä automaattisesti – erityisesti, jos 1-taso on mallinnettu yhtenä pääsekvenssinä. Jos prosessissa kuitenkin sattuu olemaan useita irrallisia ketjuja tai haarautuvia 1-tasolla olevia aloitus/lopetuspisteitä eikä prosessia voida esim. pilkkoa useampaan aliprosessiin, alku- ja loppuelementit on merkittävä eksplisiittisesti *Alku/loppu*-elementeillä. Lisäksi, jos prosessiin voidaan tulla tai lähteä kesken sen kulun (eikä kyseessä ole aliprosessi), mahdolliset liittymäpisteet merkitään *Alku/Loppu*-elementeillä. Varsinaisten aloitus- ja lopetuspisteiden lisäksi prosessiin tulevat ja lähtevät kontrolli- ja tietovirrät kiinnitetään näihin liittospisteisiin (katso luku 2.4).

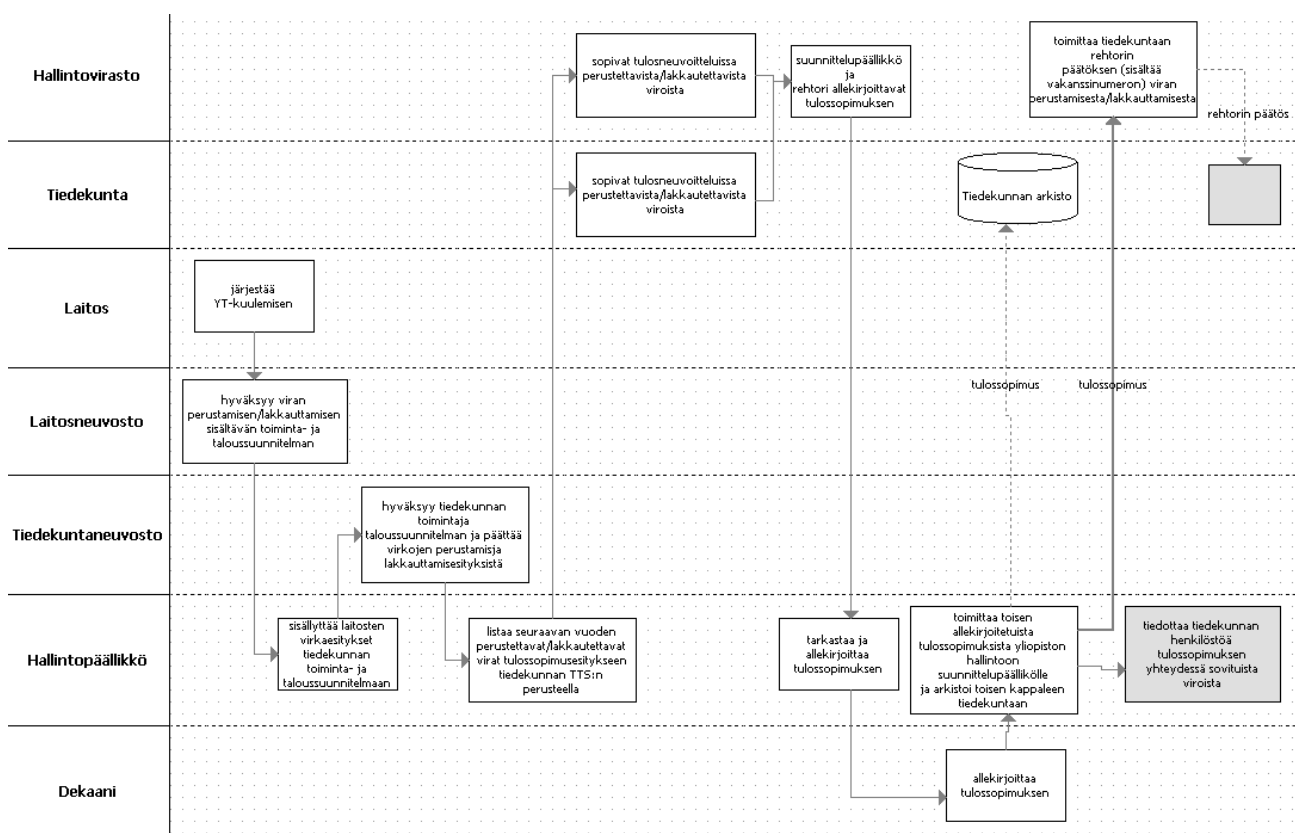
Jos prosessin työnkulussa esiintyy dokumentteja, ne tulisi ensi sijassa merkitä kontrolli- tai tietovirran dokumenteiksi. Mitään erillisiä dokumenttisympoleja ei pidä käyttää, koska niiden tulkinta muiden prosessielementtien suhteen ei ole selkeä. Jos taas dokumentti on kontrolli- tai tietovirran osana, tulkitaan automaattisesti, että dokumentti liittyy sekä edeltävään että seuraavaan mallinnuselementtiin eikä sitä tarvitse enää niihin erikseen lisätä. Merkintä dokumenttina helpottaa myös tietojen haettavuutta.

Kontrollivirran tulkinta on, että edeltävän askeleen suorituksen on päätyttävä, ennen kuin voidaan siirtyä seuraavaan. Tietovirralla vastaavaa vaatimusta ei ole. Jos kontrollivirtaan kuuluu informaatioyksikkö, käytetään *Kontrolli- ja tietovirta* -symbolia (QPR:ssä dokumentteja on mahdollista liittää myös suoraan kontrollivirtaan (*Properties*-dialogi, *Information items*-välilehti), mutta tällöin dokumentin nimi ei näy suoraan kaaviossa).



Kuva 5: WWW-sivujen uudistusprosessi tuotuna ProcMineriin.

Jos tiettyyn prosessiaskeleeseen liittyy monta toimijaa, se on merkittävä tekemällä elementit jokaiselle toimijalle niin, että elementteihin tulevilla kontrolli- tai tietovirroilla on sama lähtöelementti ja lähtevillä virroilla sama tuloelementti. ProcMiner tulkitsee tämän yhdistetyksi (*Composite*) elementiksi, jolla on monta toimijaa. Raportin luontia selkeyttää, jos kaikilla askeleilla on sama kuvaus, mutta tämä ei ole pakollista. Jokaisen *Compositeen* kuuluvan elementin on oltava samalla abstraktiotasolla. Yhdistettyjä elementtejä ei saa olla useita välittömästi peräkkäin. **Monen toimijan askelta ei saa kuvata esim. vetämällä yhtä mallinnuselementtiä monen uimaradan yli** – QPR:n tietomallissa tällöin vain ylin uimarata tulkitaan aktiiviseksi toimijaksi, vaikka visuaalinen esitys väittää muuta. Yleisesti monen prosessiaskeleen ”haarautuminen” moneen toimintoketjuun yhdestä toimintoaskeleesta viittaa rinnakkaiseen toimintaan. Jos haarautumista edeltää päätösaskel, toimintoketjut tulkitaan vaihtoehtoiksi.



Kuva 6: Viran perustaminen/lakkauttaminen TTS:n yhteydessä.

Kuvassa 6 on esimerkkejä muokatuista mallinnuselementeistä. Sivun 9 taulukossa on sama prosessi ProcMinerin generoimassa tekstimuodossa esitettyinä. Prosessin 6. askeleen symbolit *Tiedekunnan ja yliopiston hallinnon edustajat sopivat tulosneuvotteluissa perustettavista/lakkautettavista viroista* tulkitaan *Composite*-elementtinä. Prosessin 10. askeleesta (*Hallintopäällikkö toimittaa...*) lähtee *tietovirta* Tiedekunnan arkistoon ja *Kontrolli- ja tietovirta* yliopiston hallintoon. Prosessin loppuun merkitty *Rehtorin päätöksen* toimittaminen tiedekuntaan on tyypillinen ”tietovirran pää”, joka merkitään alemmalle abstraktiotasolle kuin pääsekvenssi. Taulukkomuotoisessa kuvauksessa on huomattava prosessin pääsekvenssin mallinnusta yhteen toimintoketjuun – myös 6. askeleen yhteistyössä tehtävä toiminto tulkitaan taulukon yhdeksi vaiheeksi.

Sekvenssi: 1			
Alkupiste	Laitos		
1.01	Laitosneuvosto	järjestää YT-kuulemisen, jossa kuullaan henkilöstöä viran perustamisesta/lakkauttamisesta	
1.02	Laitosneuvosto	hyväksyy viran perustamisen/lakkauttamisen sisältävän toiminta- ja taloussuunnitelman	
1.03	Hallintopäällikkö	sisällyttää laitosten virkaesitykset tiedekunnan toiminta- ja taloussuunnitelmaan	
1.04	Tiedekuntaneuvosto	hyväksyy tiedekunnan toiminta- ja taloussuunnitelman ja päättää virkojen perustamis- ja lakkauttamisesityksistä	
1.05	Hallintopäällikkö	listaa seuraavan vuoden perustettavat/lakkautettavat virat tulossopimusesitykseen tiedekunnan TTS:n perusteella	
1.06	Yliopiston hallinto Tiedekunta	Tiedekunnan ja yliopiston hallinnon edustajat sopivat tulosneuvotte- luissa perustettavista/lakkautettavista viroista	
1.07	Yliopiston hallinto	Suunnittelupäällikkö ja rehtori allekirjoittavat tulossopimuksen	
1.08	Hallintopäällikkö	tarkastaa ja allekirjoittaa tulossopimuksen	
1.09	Dekaani	allekirjoittaa tulossopimuksen	
1.10	Hallintopäällikkö	toimittaa toisen allekirjoitetuista tulossopimuksista yliopiston hallin- toon suunnittelupäällikölle ja arkistoi toisen kappaleen tiedekuntaan	tulossopimus
1.11	Yliopiston hallinto	Yliopiston hallinto toimittaa tiedekuntaan rehtorin päätöksen (sisältää vakanssinumeron) viran perustamisesta/lakkauttamisesta	tulossopimus
Päätepiste	Yliopiston hallinto		
Seuraavat sekvenssit: 1.10a, 1.10b, 1.11a			

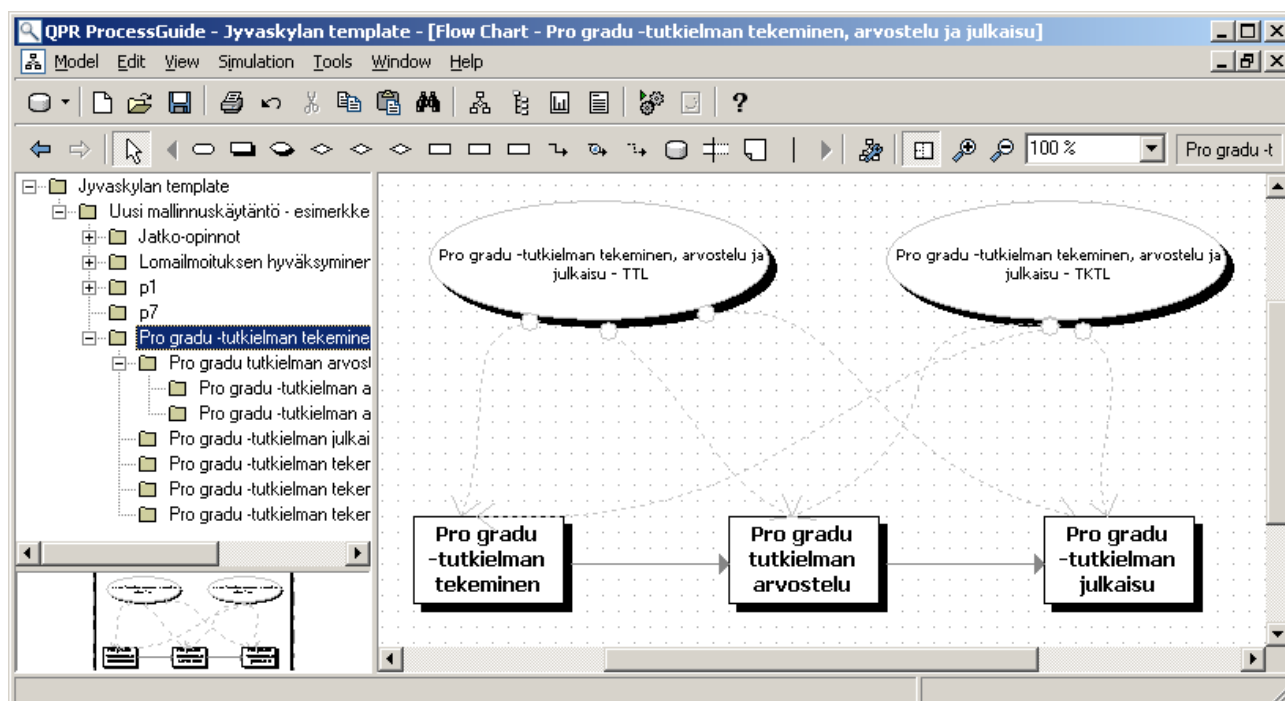
Sekvenssi 1.10a Edeltävät sekvenssit: 1			
1.10a.1	Hallintopäällikkö	tiedottaa tiedekunnan henkilöstöä tulossopimuksen yhteydessä sovitusta viroista	

Sekvenssi 1.10b Esiehto: tulossopimus Edeltävät sekvenssit: 1			
1.10b.1			tulossopimus

Sekvenssi 1.11a Esiehto: Rehtorin päätös Edeltävät sekvenssit: 1			
1.11a.1	TIEDEKUNTA		

2.4 Prosessihierarkia

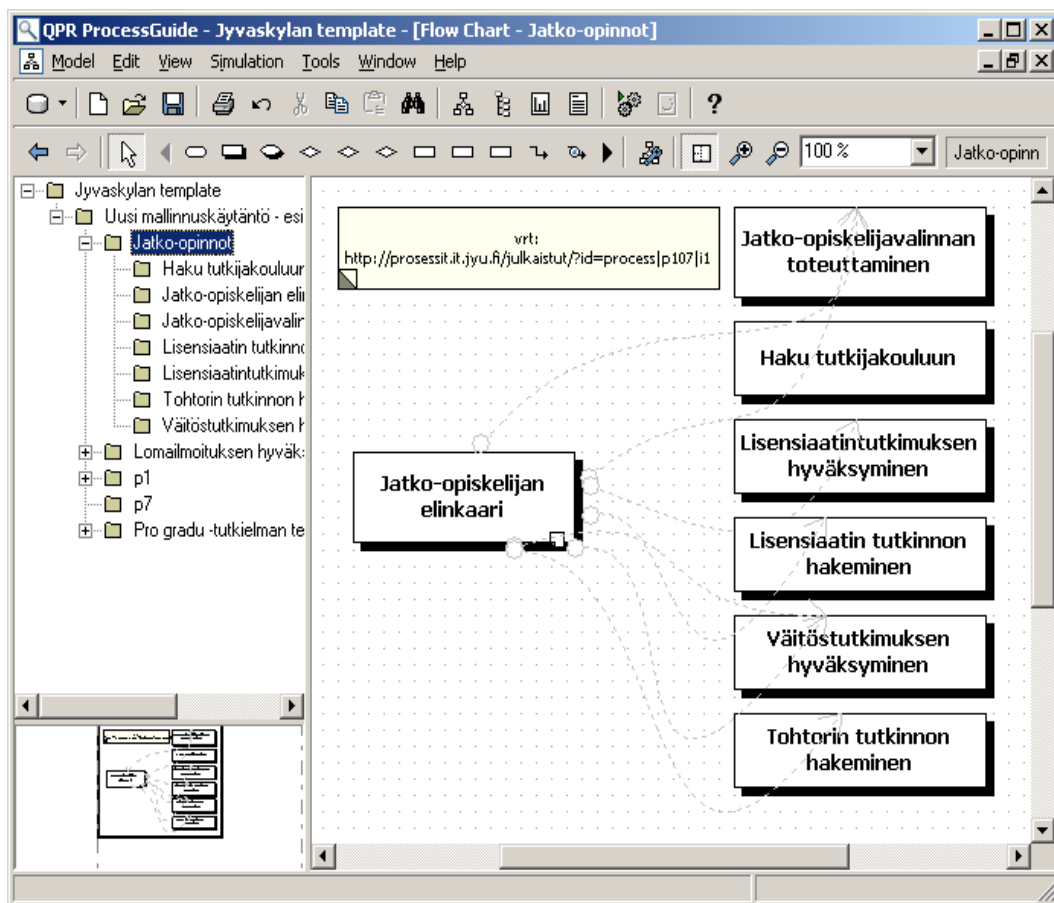
Prosessit järjestetään hierarkiaksi *Prosessi*-elementtejä käyttäen. Jos prosessista on useamman organisaation toteutuksia, toteutukset lisätään prosessin välittömiksi aliprosesseiksi *Prosessi-instanssi*-elementillä. Jos prosessilla on ”standardoitu” työnkulku, se merkitään suoraan ylempään prosessiin. Muussa tapauksessa (esim. eri laitoksilla omat toimintatavat, joista yksikään ei ole määräävä) jokainen työnkulku sijoitetaan omaan aliprosessiinsa. Tätä on havainnollistettu kuvassa 7, jossa on esitetty *Pro gradu -tutkielman tekeminen, arvostelu ja julkaisu* -prosessi. Prosessilla on kaksi vaihtoehtoista toteutusta (TTL ja TKTL), jotka on merkitty ellipseillä. Lisäksi aliprosessilla *Pro gradu -tutkielman arvostelu* on omat laitoskohtaiset instanssit (kuva 10).



Kuva 7: *Pro gradu -tutkielman tekeminen, arvostelu ja julkaisu*.

ProcMiner kopioi varsinaisessa prosessissa olevat metatiedot (omistaja, asiakas, esiehdot, jälkiehdot jne.) sellaisenaan prosessi-instansseille, jos niitä ei ole muutettu. Jokaisella prosessi-instanssilla tulisi olla kuitenkin määriteltynä ainakin instanssin toteuttava organisaatio-metatietokenttä (*Properties*-dialogin Custom Attributes -välilehti). Luettavuutta parantaa toteuttajan merkintä jollakin tavalla prosessi-instanssin nimeen. Esim. *Pro gradu -tutkielman tekeminen, arvostelu ja julkaisu – TTL*. (vrt. roolihierarkian merkintätapa luvussa 2.2).

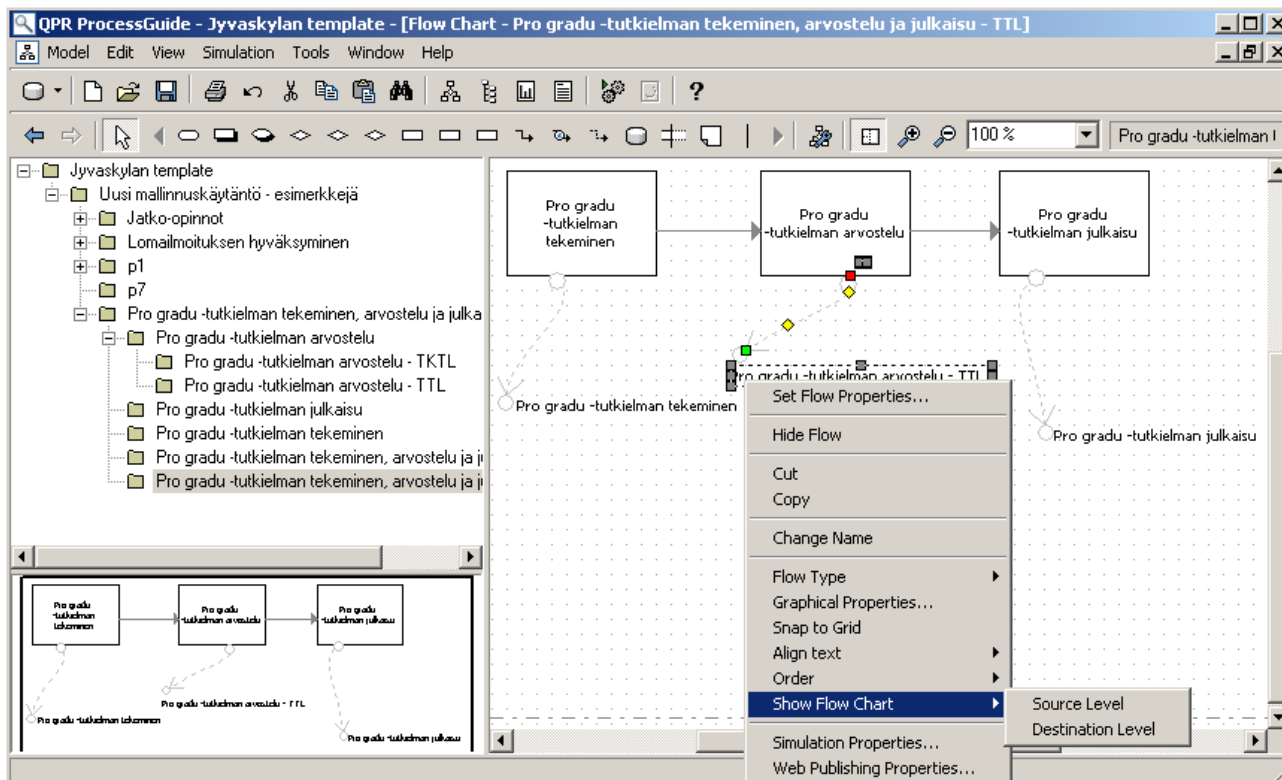
Kuvan 7 prosessilla on aliprosessit *Pro gradu -tutkielman tekeminen*, *Pro gradu -tutkielman arvostelu* ja *Pro gradu -tutkielman julkaisu*. Varsinaisten työnkulkuun liittyvien aliprosessien lisäksi aliprosesseja voidaan käyttää myös ryhmittelemään tietyille alueelle kuuluvia prosesseja. Esimerkkinä tästä ovat jatko-opiskeluun liittyvät prosessit (kuva 8). Jokaisella prosessilla voi olla edelleen omia instanssejaan tai aliprosesseja.



Kuva 8: Jatko-opiskeluun liittyvät prosessit.

Useimmissa tapauksissa aliprosesseihin viittaukseen ei voida käyttää QPR:n omia aliprosessisymboleja, vaan viittaus on tehtävä erillisellä *Viite aliprosessiin* -nuolella. Tällaisia tilanteita ovat ainakin seuraavat:

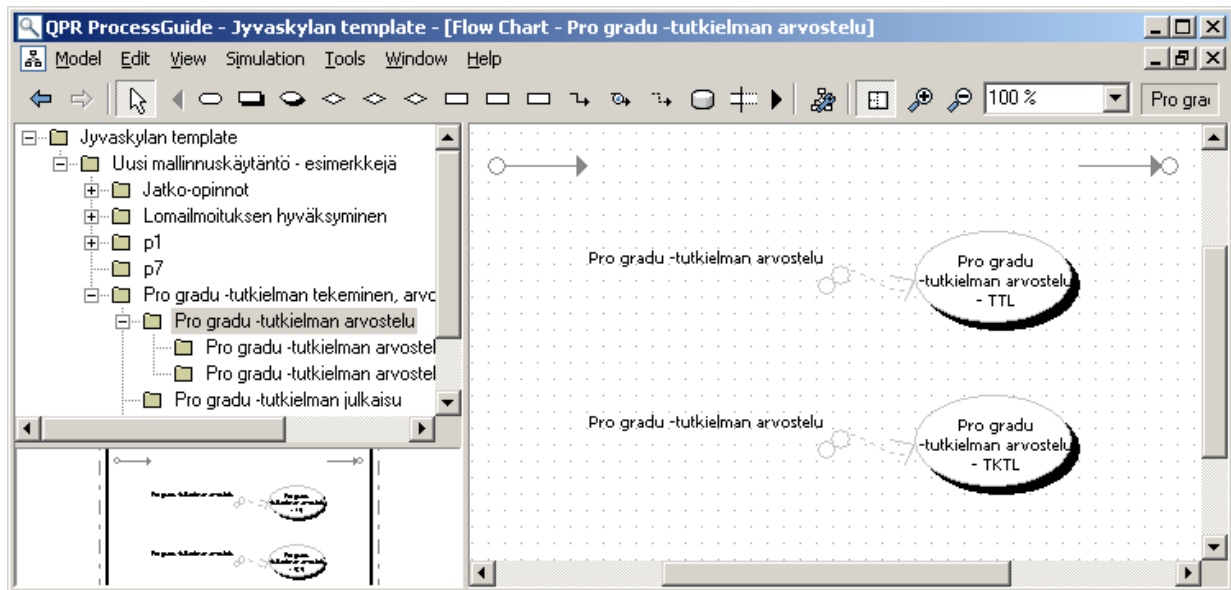
- Tietyn prosessin instanssista viitataan varsinaisen prosessin johonkin aliprosessiin (Esim. *Pro gradu tutkielman tekeminen, arvostelu ja julkaisu – TTL* -prosessista viitataan prosessiin *Pro gradu -tutkielman tekeminen*.)
- Tietyn prosessin instanssista viitataan varsinaisen prosessin jonkin aliprosessin instanssiin (Esim. *Pro gradu tutkielman tekeminen, arvostelu ja julkaisu – TTL* -prosessista viitataan prosessi-instanssiin *Pro gradu -tutkielman arvostelu – TTL*.)
- Yleisesti: tietyistä prosessista tai prosessi-instanssista viitataan toisaalla määriteltyyn prosessiin tai prosessi-instanssiin. Tällöin kyseessä on aidosti ”jaettu” aliprosessi, joita QPR:llä ei voi suoraan mallintaa.



Kuva 9: Aliprosessiviittaus.

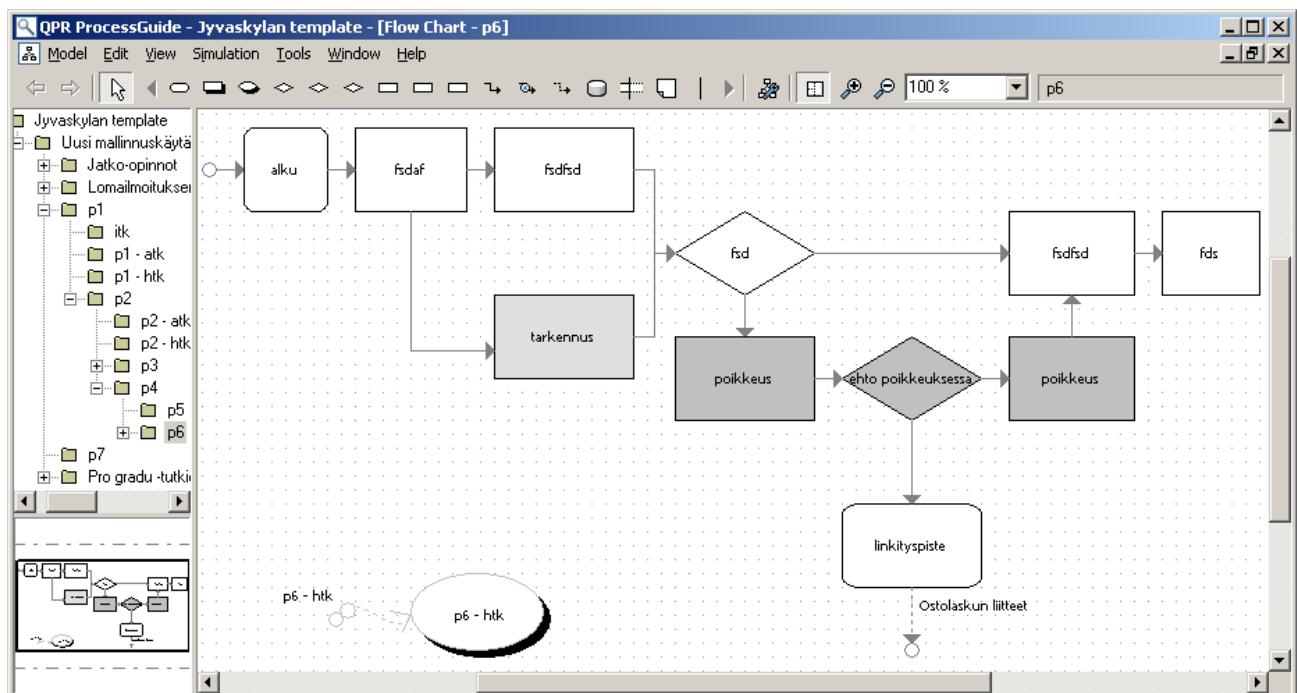
Kuvassa 9 on esitetty aliprosessiviittaukset prosessi-instanssista *Pro gradu tutkielman tekeminen, arvostelu ja julkaisu – TTL*. Kohteena oleva aliprosessi määritetään nuolen ponnahdusvalikon *Set Flow Properties...* -dialogilla. Prosessien välillä voidaan navigoida ponnahdusvalikon *Show Flow Chart* -alimenulla. Viitteen määrittelyn jälkeen QPR generoi nuolen näkyviin muissakin prosesseissa vastaavaan kohtaan. Esimerkiksi kuvissa 7 ja 8 näkyvät vaaleanharmaat nuolet on alun perin määritelty toisaalla. Generoitujen nuolien asetteluun ei kannata kiinnittää suurta huomiota, koska niiden olemassaolo riippuu alkuperäisestä aliprosessiviitteestä eikä niiden asettelu ole siirrettävissä ProcMineriin helposti. **Generoitujen aliprosessinuolet päitä ei saa irrottaa, koska tällöin aliprosessilinkki katoaa myös alkuperäisestä kohdasta.**

QPR generoi myös linkkejä prosessien välille, jos niitä on määritelty yliprosessissa. Esimerkiksi (kuva 7) *Pro gradu -tutkielman arvostelu* -prosessiin on tuleva linkki *Pro gradu -tutkielman tekemisestä* ja lähtevä linkki *Pro gradu -tutkielman julkaisuun*. Tämä näkyy aliprosessissa (kuva 10) vasemmassa ja oikeassa yläkulmassa olevina nuolina (huom. prosessi-instansseihin viittaavat aliprosessinuolet ovat myös generoituja – toinen TTL:n ja toinen vastaavasti TKTL:n graduprosessi-instanssista). Prosessille ei ole määritelty oletusarvoista työnkulkua – vain instanssit). **Aliprosessinuolet tapaamaan myöskään generoituja linkkejä ei saa poistaa, koska tällöin kontrollivirta katoaa myös yliprosessista.** Generoidut linkit saa liittää kohdeprosessissa vastaaviin *Alku/Loppu* -symboleihin (jos sellainen on – esim. kuvassa 10 ei ole, mutta kuvassa 11 on alkusymboli), mutta tämä ei ole pakollista, koska linkin liitoskohta on useimmissa tapauksissa pääteltävissä automaattisesti.

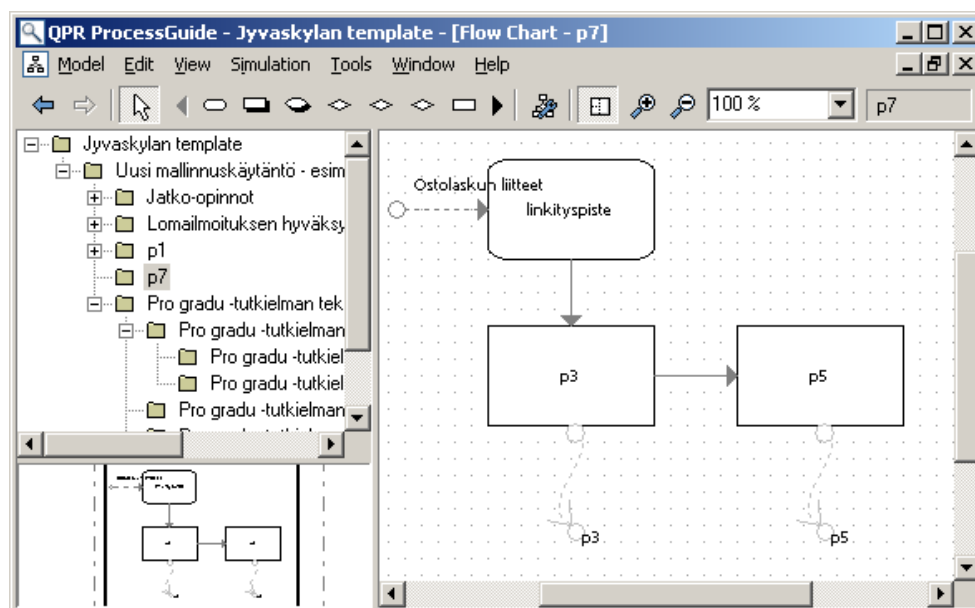


Kuva 10: Generoidut prosessien väliset linkkinuolet.

Jos prosessien välillä on yhteyksiä, joita ei voida esittää QPR:n normaalilla aliprosessikäytännöllä, aliprosessinuolilla tai generoiduilla prosessien välisillä linkeillä, prosessien välinen linkki on tehtävä manuaalisesti. Yleensä tarvetta linkitykselle on, jos tietoa halutaan tuoda tai viedä prosessista toiseen kesken prosessin kulun. Linkitys voidaan tehdä joko tavanomaisella kontrolloi- tai tietovirralla, mutta muunnoksen helpottamiseksi linkityspisteinä tulisi aina käyttää *Alku/Loppu*-symbolia (pyöristetty suorakulmio). Linkityksen kohdeprosessi voidaan määrittää samalla *Set Flow Properties* -dialogilla kuin aliprosessiviittauksessa. Linkitystä on havainnollistettu kuvissa 11 ja 12, missä prosessien välillä on *Ostoslaskun liitteet* -tietovirta. **Prosessien välistä linkkiä tehtäessä tietovirran molempien päiden tulee olla kiinnittyneenä linkityspisteeseen.** Muuten lähde- ja kohdeprosesseja ei voida tunnistaa. Erityisesti pelkkä prosessin nimen kirjoittaminen ei riitä (eikä ole edes tarpeellista, jos linkitys tehdään oikein – prosessin nimi tulee tällöin tietovirtaan näkyviin automaattisesti).



Kuva 11: Prosessien välisen linkin (Ostolaskun liitteet) lähtöprosessi.



Kuva 12: Prosessien välisen linkin (Ostolaskun liitteet) kohdeprosessi.

3 Tekninen toteutus

Luvussa kuvataan tietomallin muutoksia ja kuvauskäytäntöjä tekniseltä kannalta.

3.1 QPR:n ja ProcMiner'in tietomallien eroista

ProcMiner'in prosessit sisältävät joukon metatietokenttiä, joilla prosesseja kuvataan (omistaja, asiakas, esiehdot, jälkiehdot jne). QPR:ssä näitä ei oletuksena ole, mutta ne voidaan lisätä ohjelman mallinnusasetusten kautta (valikon *Tools -> Modeling Options*). Metatietokenttien avulla prosessikuvaukseen liittyvät tekstikuvaukset saadaan pidettyä yhdessä paikassa (vrt. Kaari-projektin tuotos, jossa metatiedot ja prosessikaaviot ovat erillisissä dokumenteissa¹).

ProcMiner mahdollistaa prosessin kuvauksen monella eri abstraktiotasolla. Tällöin prosessin oletusarvoinen ja ”tyypillisin” kulku, tarkentavat askeleet ja lopulta poikkeukset voidaan merkitä omille tasoilleen. Tämä helpottaa oleellisesti prosessin kulun hahmottamista, ja prosessista voidaan generoida kohdennettuja raportteja, joissa näkyvät vain valitulla tarkkuudella olevat aktiviteetit.

ProcMiner erottelee organisaatiot muista rooleista (esim. virkanimikkeet), jotka voivat edelleen kuulua tiettyyn organisaatioon. Tämä mahdollistaa roolien raportoinnin eri tarkkuuksilla katsojan näkökulmasta riippuen (esim. uimaradan nimenä *tiedekunta* tai *dekaani*). Aiemmassa QPR-mallissa tämä täytyi ilmaista keinotekoisilla ”Tiedekunta / Dekaanin”-tyylisillä nimeämiskäytännöillä, joiden täsmällinen tulkinta ei ole selkeä.

QPR:n prosessihierarkiassa jokainen prosessi voi olla korkeintaan yhden prosessin aliprosessina. ProcMinerissa prosessilla voi olla useita yliprosesseja. Tämä mahdollistaa aliprosessien uudelleen käytön, jos tietty prosessin osa toistuu samana useissa eri prosesseissa, tai eri organisaatioissa prosessin osa toistuu samalla tavalla.

ProcMiner erottelee prosessin ja prosessi-instanssin käsitteet. Prosessi kuvaa tietyn vakiintuneen toimintatavan yleiset ominaisuudet ja mahdollisesti oletusarvoisen työnkulun, prosessi-instanssi on tietyn yksikön toteutus prosessista. Prosessi-instanssit mahdollistavat monen eri organisaation prosessien sisällyttämisen samaan malliin niin, että mahdollisimman suuri osa yhteisistä tiedoista on yhdessä paikassa. QPR:ssä aliprosesseja voidaan oletuksena mallintaa symboleilla *Ydinprosessi* ja *aliprosessi*, mutta näiden erona on lähinnä tapa merkitä prosessihierarkian huipulla olevat prosessit ydinprosessin nimellä. Siksi muokatussa mallissa on tavallisten aliprosessien merkintää varten käytössä vain jälkimmäinen symboli.

3.2 Muutokset QPR-malliin

JY:n aiempi mallinnuspohja sisältää muutamia tarpeettomia elementtejä, jotka voidaan poistaa tai korvata muilla elementeillä – samalla QPR:n käyttöliittymä selkiytyy, koska elementit häviävät myös ohjelman työkalupalkista. QPR-tietomallia voi muuttaa valikon *Tools -> Modeling Options* -dialogilla. Poistettavia elementtejä ovat:

- *Aliprosessi* uudelleennimetään elementiksi *Prosessi*.
- *Subprocess*. Korvataan elementillä *Prosessi*.
- *Ydinprosessi*. Korvataan elementillä *Prosessi*.
- *Activity*. Korvataan elementillä *Toiminto*.

¹<http://projekti.it.jyu.fi/2004/Kaari/tuotos/dokumentit/kaari-tuotos.pdf>

- *Decision*. Korvataan elementillä *Päätös*.
- *Dokumentti*. Korvataan elementillä *Toiminto*. Näkyvän dokumenttisyömbolin sijaan käytetään tietovirtoihin liittyviä informaatioyksiköitä.
- *Ulkoinen malli*. Poistetaan, ei ole käytössä missään.
- *Materiaalivirta*. Poistetaan. On käytössä muutamissa (talous)hallinnon prosesseissa, mutta virran pitäisi useimmissa tapauksissa olla kontrollivirta.
- *Material store*. Poistetaan, ei ole käytössä missään.
- *Group*. Poistetaan, ei ole käytössä missään.

Seuraavat uudet mallinnuselementit lisätään:

- *Prosessi-instanssi*. Ellipsi.
- *Toiminto (2 – tarkentava)*. Suorakulmio, esim. vaaleanharmaa väri.
- *Toiminto (3 – poikkeus)*. Suorakulmio, esim. tummanharmaa väri.
- *Päätös (2 – tarkentava)*. ”Salmiakki”, esim. vaaleanharmaa väri.
- *Päätös (3 – poikkeus)*. ”Salmiakki”, esim. tummanharmaa väri.
- *Kontrollivirta Viite aliprosessiin*. Katkoviiva, muita nuolia vaaleampi väri.
- *Tietovirta Kontrolli- ja tietovirta*. Paksu, yhtenäinen viiva.

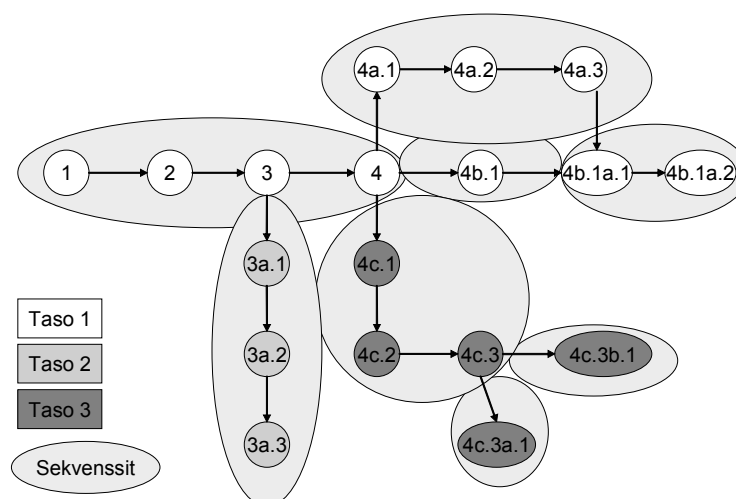
Mallinnuselementeille lisätään seuraavat metatietokentät:

<i>Elementti</i>	<i>Kentän nimi ja kohdetyyppi</i>
Prosessi	Asiakas (organisaatioyksikkö) Tarkoitus (monirivinen teksti) Esiehdot (teksti) Jälkiehdot (teksti) Syötteet (monirivinen teksti) Tuotokset (monirivinen teksti) Delegointitiedot (monirivinen teksti) Asiakasvaatimukset (monirivinen teksti) Menestystekijät (monirivinen teksti) Rajoitteet (monirivinen teksti) Kehitysehdotukset (monirivinen teksti) Ongelmat (monirivinen teksti) URL (teksti)

Prosessi-instanssi	Samat metatietokentät kuin <i>Prosessilla</i> . Lisäksi Toteuttajaorganisaatio (Organisaatioyksikkö) Poikkeukset (monirivinen teksti)
Päätös (kaikki tasot)	URL (teksti)
Toiminto (kaikki tasot)	URL (teksti)
Informaatioyksikkö	URL (teksti)
Organisaatioyksikkö	URL (teksti) Organisaatioyksikkö (Organisaatioyksikkö)
Tietojärjestelmä	URL (teksti)

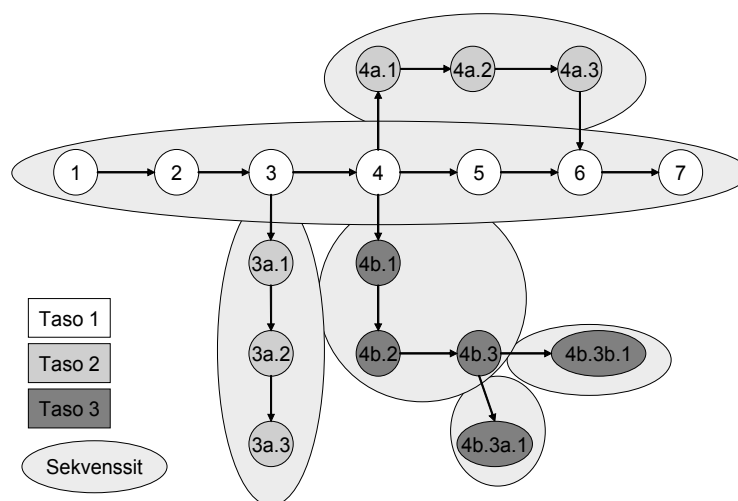
3.3 Prosessin abstraktiotasot ja sekvenssit

Sisäisesti ProcMiner tulkitsee prosessin työnkulun monitasoisena, ositettuna graafina, joka ositetaan samalla abstraktiotasolla olevien ketjujen mukaan. Ketjujen avulla prosessin elementit voidaan numeroida ja järjestää yksikäsitteisesti ja kaavion lisäksi voidaan muodostaa tekstimuotoinen kuvaus. Tämä saattaa olla kaaviomuotoa selkeämpi hahmottaa, jos prosessi on pystytty osittamaan mahdollisimman pieneen määrään ketjuja. Tästä syystä prosessista tulisi aina jo mallinnusvaiheessa pyrkiä hahmottamaan ketjumaisia rakenteita.



Kuva 13: ProcMiner-graafin ositus.

Esimerkki ”väärin” mallinnetusta prosessista on kuvassa 13, koska 1-tason elementit eivät muodosta yhtä ketjua. Tämän seurauksena 1-taso jakautuu neljään erilliseen ketjuun. Parempi tapa on esitetty kuvassa 14, jossa pääsekvenssiä sekoittavat kolme prosessiaskelta on siirretty tasolle 2.

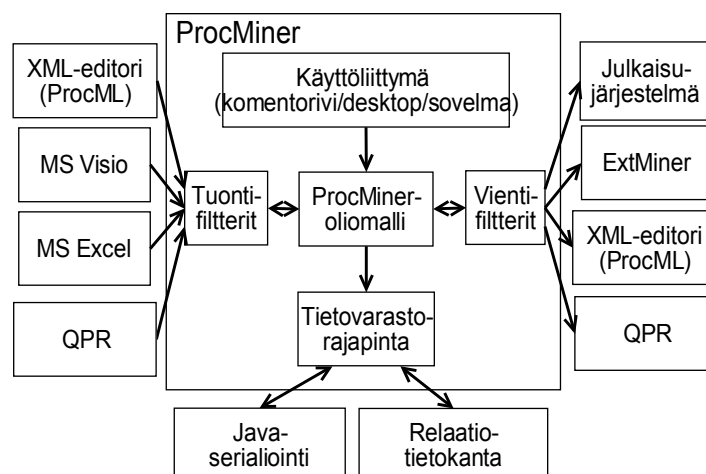


Kuva 14: ProcMiner-graafin ositus - selkeytetty 1-taso.

Jos tiettyyn prosessiaskeleeseen liittyy monta toimijaa, se on merkittävä tekemällä rinnakkain elementti jokaiselle toimijalle niin, että elementteihin tulevilla kontrolli- tai tietovirroilla on sama lähtöelementti ja lähtevillä virroilla sama tuloelementti. ProcMiner tulkitsee tämän yhdistetyksi (*Composite*) elementiksi, jolla on monta toimijaa, eikä pura toimintoketjua (vrt. kuva 14) tältä osin. Yhdistettyjen elementtien avulla voidaan kiertää useimpien prosessimallinnusvälineiden sallima korkeintaan yhden roolin liittäminen prosessiaskeleeseen, säilyttäen samalla mallin siirrettävyys.

3.4 Integroinnin laajuus ja versionhallinta

1-suuntainen integrointi QPR:stä ProcMineriin on suhteellisen helppoa toteuttaa (prototyypitasoinen toiminto, joka mm. pystyy lukemaan JY:n hallinnon ja johtamisen prosessit on jo toteutettu), mutta 2-suuntainen muunnos, joka huomioi monta samanaikaista käyttäjää on huomattavasti vaativampi. Muunnos kannattaa toteuttaa vaiheittain, jolloin osa integroinnin eduista on välittömästi käytettävissä. ProcMinerin filteripohjainen arkkitehtuuri (katso kuva 15) mahdollistaa myös muiden ohjelmien käytön prosessien kuvauksessa, mutta täysi ilmaisuvoima on toistaiseksi vain XML-muotoisella ProcML-kielellä.



Kuva 15: ProcMiner-arkkitehtuuri.

2-suuntaisesta integroinnista voidaan erottaa seuraavat vaiheet toteutuksessa vaadittavan työmäärän laajuuden mukaan.

1. **Mallitietokannan synkronointi.** Rooli- ja dokumenttihierarkian ja tietojärjestelmälistan siirto järjestelmien välillä (vrt. luku 2.2)
2. **Yksittäisten prosessimallien synkronointi.** Käsittää ensimmäisen kohdan lisäksi yhden prosessimallin viennin QPR:ään kerrallaan. Prosessien välisiä linkkejä, prosessi-instansseja tai aliprosesseja ei huomioida (luku 2.3).
3. **Prosessihierarkian synkronointi.** Täysi 2-suuntainen prosessimallien integrointi käsittäen koko prosessihierarkian linkkeineen (luku 2.4).

Riippumatta siitä, missä laajuudessa integrointi toteutetaan, QPR-malli on säilytettävä hallitusti yhdessä tietovarastossa, josta eri laitokset voivat muokata omia osiaan. Edellytyksenä on versionhallintajärjestelmä (esim. Subversion, CVS), jonka kautta käyttäjät pyytävät QPR- tai ProcMiner-muotoista mallia omalle koneelleen, suorittavat muutokset ja vievät mallin (tai pelkästään muuttuneet osat) takaisin järjestelmään ProcMinerin tai mahdollisesti QPR:n oman prosessiportaalin kautta, joka raportoi mahdollisista konfliktitilanteista (esim. joku toinen on muuttanut samaa prosessia tai muuttanut roolihierarkiaa prosessissa olevien roolien osalta). Versiohallintajärjestelmän ansiosta kuvausten muutoshistoria on saatavilla ja virhetilanteissa voidaan aina palata aiempaan versioon. Valitusta tekniikasta riippumatta eri laitosten prosessimallien yhteensovittaminen sekä rooli- ja dokumenttihierarkioiden koordinointi edellyttää selkeitä käytäntöjä ja yhteistyötä.

4 Yhteenveto

ProcMiner-sovelluksella on IT-tiedekunnassa mallinnettu vuosien 2005-2006 aikana yli 150 erita-soista prosessikuvausta keskittyen tiedekunnan ja laitosten henkilöstö- ja opintohallinnon prosesseihin sekä prosessien hallinnan ja mallinnuksen käytäntöihin. Erityisesti henkilöstöhallinnon kuvausten osalta suuri osa kuvauksista on todennäköisesti samoja muiden laitosten vastaavien prosessien kanssa, joten prosessien uudelleenkäyttö helpottaisi JY:n laatuhankekokonaisuutena. Jyväskylän yliopistossa käyttöönotetulla QPR-sovelluksella on mallinnettu pääasiassa yliopiston hallinnon prosesseja, jotka ovat luettavissa ProcMineriin, mutta malleissa on erityisesti rooli- ja dokumenttihierarkioiden osalta epätarkkuuksia, mikä vaikeuttaa niiden suoraa linkitystä muihin prosesseihin. QPR:n ja ProcMinerin lisäksi jotkut laitokset ovat mallintaneet prosesseja ainakin Microsoft Visiol-la.

Sovellusten tietomalleissa on oleellisia eroja, mutta lisäämällä QPR-puolelle uusia mallinnuselementtejä ja tarkentamalla mallinnuskäytäntöjä on mahdollista toteuttaa 2-suuntainen, tiedot säilyttävä muunnoskomponentti. Muunnos kannattaa toteuttaa vaiheittain, jolloin jolloin osa integroinnin eduista on välittömästi käytettävissä eikä mallintajan tarvitse omaksua laajahkoa tietomallia kerralla. Toimiva muunnos mahdollistaa paitsi XML-muotoisten ProcML-kuvausten, myös tietyin rajoituksin Visiota tai Exceliä käyttävien prosessimallien viennin ProcMinerin kautta QPR:ään. ProcMineria on mahdollista hyödyntää rakenteisessa tiedonhaussa ja prosessien laskennallisessa vertailussa tiedonlouhintaa käyttäen. Lisäksi ProcMiner sisältää julkaisujärjestelmän, joka luo selättävän html-sivuston ja pdf-muotoiset prosessikäsikirjat. Uudet mallinnuselementit vaativat hieman aiempaa enemmän mallinnustyötä, mutta toisaalta prosessikuvausten haettavuus, ymmärrettävyys ja ylläpidettävyys paranevat. Prosessien rakenteen tai esim. roolitusten muuttuessa prosessikuvausten ylläpitoon on kiinnitettävä erityistä huomiota, jotta prosessimallit kuvaavat organisaation todellista toimintaa ja ohjaavat osaltaan toiminnan jatkuvaa kehittämistä.