ROOLIPOHJAINEN PÄÄSYNHALLINTA

Roolien määrittelymenetelmät

Janne Kallunki
Pro gradu -tutkielma
Tietojenkäsittelytiede
Itä-Suomen yliopiston
tietojenkäsittelytieteen laitos
Toukokuu 2012

ITÄ-SUOMEN YLIOPISTO, Luonnontieteiden ja metsätieteiden tiedekunta, Kuopio
Tietojenkäsittelytieteen laitos
Tietojenkäsittelytiede

KALLUNKI JANNE, P.: Roolipohjainen pääsynhallinta. Roolien määrittelymenetelmät.
Pro gradu -tutkielma, 63 s., 2 liitettä (2 s.)
Pro gradu -tutkielman ohjaajat: YTM, LitM Taina Kurki ja FT Matti Nykänen
Toukokuu 2012

Avainsanat: RBAC, role-based access control, role engineering, roolit, pääsynhallinta

Tämän tutkielman tarkoituksena oli perehtyä roolipohjaisen pääsynhallinnan toimintaan (RBAC) ja erityisesti siihen kiinteästi kuuluvaan roolien määrittelyvaiheeseen. Roolipohjainen pääsynhallinta on kasvattanut suosiotaan yritysmaailmassa sen tuoman hallinnollisten ja taloudellisten etujen johdosta. Roolipohjaisen pääsynhallinnan toteutuksen kallein ja työläin vaihe on roolien määrittely. Tästä syystä sopivan menetelmän valitseminen roolien määritykseen on tärkeää. Tutkielman teoriaosa on toteutettu käyttäen perinteistä kirjallisuuskatsausta.

Roolien määrittelyn tutkimusta lähestyttiin systemaattisen kirjallisuuskatsauksen menetelmin. Aihetta käsitteleviä artikkeleita haettiin sähköisistä tietokannoista. Käytetyt tietokannat olivat: ACM, IEEE Xplore, ScienceDirect, Web of Science ja CiteSeerX. Aineistoksi valikoitui kahdeksan tutkimusta: Coynen perusmenetelmä, käyttötapauksiin perustuva menetelmä, komponenttiteknologiaa hyödyntävä menetelmä, prosessikeskeinen menetelmä, roolin elinkaaren perustuva menetelmä, skenaarioperusteinen menetelmä, tavoiteperusteinen menetelmä ja integroitu menetelmä.

Tutkimuksen tuloksena saatiin selville, että roolien eri määrittelymenetelmät lähestyivät ongelmaa hyvin erilaisista suunnista. Osa menetelmistä oli keskeneräisiä ja osa oli kuvattu vain hyvin karkealla tasolla. Käyttökelpoisimmat menetelmät olivat skenaarioperusteinen menetelmä ja integroiva menetelmä. Johtopäätöksenä voidaan todeta, että sopivan menetelmän valinnalla voidaan vaikuttaa suotuisasti roolien määrittelyprosessin lopputulokseen. Alaa vaivaa kuitenkin tutkimustiedon puute, joka tuli esille aineiston hankinnassa sekä tutkijoiden itsensä lausumana.

UNIVERSITY OF EASTERN FINLAND, Faculty of Science and Forestry, Kuopio
School of Computing
Computer Science

KALLUNKI JANNE, P.: Role Based Access Control. Role engineering methods.
Master's Thesis, 63 p., 2 appendix (2 p.)
Supervisors of the Master's Thesis: MSocSc, MSc(SportsScience) Taina Kurki and PhD Matti Nykänen
May 2012

Keywords: RBAC, role-based access control, role engineering, roles, access control

The purpose of this master's thesis was to study Role-Based Access Control (RBAC) and in particular, the role engineering methods found in the scientific literature. RBAC has gained popularity in business world, because of administrative and economical advantages it brings. The most expensive and laborious phase in implementing a complete RBAC-system is role engineering. Choosing the appropriate way to define roles is therefore very important. The theoretical part of this thesis has been written using the traditional literature review.

The role engineering methods found in the scientific literature were evaluated by the means of systematic literature review. The articles used in the survey were searched using electronic databases, which were: ACM, IEEE Xplore, ScienceDirect, Web of Science and CiteSeerX. The final review consisted of eight studies: Coyne's basic method, Use Cases approach, component-based approach, process-oriented approach, role life-cycle approach, scenario-driven approach, goal-driven approach and integrated approach.

The result of the study showed that studied methods for defining roles varied significantly. Some were incomplete and some were described only in coarse-grained level. Most useful and adequate methods were scenario-driven and integrated approaches. It can be concluded that choosing a suitable method could have a positive impact on the result of the role engineering process. However, there seems to be lack of research in this field and further research effort is needed. This was realized in lack of research articles and was also mentioned by the researchers themselves in their articles.

Esipuhe

Tämä tutkielma on tehty Itä-Suomen yliopiston tietojenkäsittelytieteen laitokselle keväällä 2012. Tutkielman ohjaajina toimivat Taina Kurki ja Matti Nykänen, joille haluan osoittaa erityiskiitoksen.

Erityiskiitokset haluan osoittaa vaimolleni ja vanhemmilleni tuesta opiskelujeni aikana.

Suuri kiitos myös lapsilleni, Allille ja Antille, jotka jaksoivat muistuttaa taukojen merkityksen tärkeydestä kirjoitusprosessin aikana ☺

Kuopiossa 30.5.2012

_____

Janne Kallunki

Käsitteet ja lyhenteet

ACL	Access Control List
CORBA	Common Object Request Broker Architecture
COM	Component Object Model
DAC	Discretionary Access Control
DCOM	Distributed Component Object Model
IDL	Interface Definition Language
MAC	Mandatory Access Control
RBAC	Role-Based Access Control

Sisällysluettelo

1 JOHDANTO

Tietojärjestelmät ovat kasvaneet entistä suuremmiksi ja monimutkaisemmiksi. Ei ole tavatonta, että yrityksissä on käytössä kymmeniä, jopa satoja eri tietojärjestelmiä. Samaan aikaan niistä on tullut entistä enemmän liiketoimintakriittisiä. Tietojärjestelmien ongelmat voivat aiheuttaa huomattavia kustannuksia ja pahimmassa tapauksessa ne voivat kaataa koko yrityksen. Tämä on luonut painetta kehittää parempia tietoturvaratkaisuja, kehittää tietojärjestelmien hallintaa, varautua alati kasvaviin tietoturvauhkiin sekä hallita kasvavia tietoturvakustannuksia.

Eräs tapa vastata kasvaviin haasteisiin on roolipohjainen pääsynhallinta, joka on saavuttanut kasvavaa suosiota yritysmaailmassa. Alun perin tieteellinen malli on levinnyt kaupalliselle puolelle standardisoinnin avustuksella. Roolipohjaisuuden suosio perustuu muun muassa sen tuomaan parempaan tietoturvaan ja hallinnointikustannusten pienenemiseen.

Tässä tutkielmassa perehdytään roolipohjaiseen pääsynhallintaan. Erityistä huomiota kiinnitetään roolien määrittelyyn, joka on yksi tärkeimmistä vaiheista toimivan roolipohjaisen järjestelmän suunnittelussa. Perinteisesti roolien määrittely on tehty ad hoc -tyyppisesti, mutta alan kirjallisuudessa on esitetty myös kehittyneempiä menetelmiä. Tutkielman tarkoituksena on etsiä näitä tieteellisessä kirjallisuudessa esiintyviä apukeinoja, menetelmiä tai prosesseja, jotka helpottavat roolien määrittelytyötä

Luvussa kaksi luodaan lyhyt katsaus pääsynhallinnan toimintaperiaatteeseen ja esitellään yleisimmät käytössä olevat perinteiset pääsynhallintamallit. Tämän jälkeen esitetään roolipohjaisen pääsynhallinnan toimintaperiaate, peruskäsitteistö ja alan standardointipyrkimykset. Luvussa tuodaan esille myös roolipohjaisuuden tuomia hyötyjä ja vastaavasti tilanteita, joissa muiden menetelmien käyttö on perusteltua. Luvun tarkoituksena on toimia johdatuksena roolipohjaiseen pääsynhallinnan erityispiirteisiin.

Luvussa kolme etsitään eri tapoja suorittaa onnistunut roolien määrittely. Roolien määrittely on työläin ja kallein vaihe toimivan roolipohjaisen pääsynhallinnan suunnittelussa ja toteutuksessa [FKC97;GGM10]. Tästä syystä eri menetelmät, jotka helpottavat tätä työtä, voivat alentaa kustannuksia, vähentää tarvittavaa työmäärää sekä parantaa loppu-

tuloksen laatua. Aihepiiristä on kirjoitettu joitain pro gradu -tasoisia opinnäytetöitä, mutta niissä ei paneuduta roolien määrittelyn ongelmaan tästä näkökulmasta. Näistä syistä johtuen roolien määrittelyn eri apukeinojen tunteminen ja vertailu on mielenkiintoinen ja hyödyllinen tutkimisen kohde. Tutkimusongelmani ovat:

1. Mitä menetelmiä on käytettävissä tukemaan roolien määrittelytyötä?
2. Kuinka tarkalla tasolla edellä esitetyt menetelmät ovat kuvattu ja kuinka käyttökelpoisia ne ovat?

Tutkimusongelmia lähestytään systemaattisen kirjallisuuskatsauksen menetelmin hakemalla, analysoimalla ja jäsentämällä aiheesta kirjoitettuja tieteellisiä artikkeleita mahdollisimman kattavasti. Menetelmän tarkempi kuvaus ja analysointi perusteet esitetään luvussa kolme vastaten ensimmäiseen tutkimusongelmaan. Luvussa neljä esitetään kirjallisuuskatsauksen tuloksien tarkastelu ja vastataan toiseen tutkimusongelmaan. Luvussa viisi pohditaan saatuja tuloksia ja esitetään mahdollisia jatkotutkimusaiheita.

2 ROOLIPOHJAINEN PÄÄSYNHALLINTA

Ferraiolo ja Kuhn [FeK92] esittivät vuonna 1992 formaalin roolipohjaisen pääsynhallintamallin. Mallin perusperiaate on se, että kaikki pääsynhallintaa vaativat toimenpiteet tapahtuvat roolien välityksellä. Tässä luvussa tarkastellaan aluksi lyhyesti pääsynhallinnan yleistä toimintaa, jonka periaatteet ja tarkoitusperät ovat voimassa kaikissa pääsynhallintamalleissa. Tämän jälkeen luodaan katsaus roolipohjaisen pääsynhallinnan toimintaperiaatteeseen sekä sen tuomiin erinäisiin hyötyihin verrattuna perinteisiin tapoihin. Luvussa esitellään myös keskeisimmät RBAC-standardit.

2.1 Pääsynhallinnan toimintamekanismi

Pääsynhallintaa on käytetty kautta aikain rajoittamaan pääsyä tärkeisiin tietoihin ja asioihin. Vartijat, portit ja lukot ovat esimerkkejä tällaisesta varhaisen ajan pääsynhallinnasta [FKC07]. Tässä tutkielmassa ei tarkastella kuitenkaan lähemmin näitä fyysisen pääsynhallinnan piiriin kuuluvia menetelmiä, vaan pääsynhallinnalla tarkoitetaan tämän tutkielman yhteydessä tietokonejärjestelmien tai niihin rinnastettavien systeemien pääsynhallintaa.

Aliluvuissa selvitetään pääsynhallinnan tarkoitusta, toimintaperiaatetta ja sen suhdetta muihin tietoturvapalveluihin. Aliluvuissa perehdytään myös pääsynhallinnan yleisempiin hallintamalleihin ja -mekanismeihin.

2.1.1 Pääsynhallinnan tarkoitus

Pääsynhallinta on yksi näkyvimmistä tietoturvamekanismeista nykypäivänä. Se on käytössä lähes joka järjestelmässä ja tästä laajuudesta johtuen se aiheuttaa myös arkkitehtuurisia ja hallinnollisia ongelmia. Liiketoiminnan kannalta ajateltuna pääsynhallinnan avulla on mahdollista jakaa resursseja optimaalisesti. Toisaalta väärin toteutettuna pääsynhallinta voi turhauttaa käyttäjiä, aiheuttaa suuria hallinnollisia kustannuksia, sallia oikeudettoman pääsyn tai arvokkaan tiedon tuhoutumisen. [FKC07] Käyttäjät voivat turhautua esimerkiksi tarvittavien käyttöoikeuksien odotteluun tai liian tiukkoihin pääsyrajoituksiin, jolloin työtehtävien hoitaminen vaikeutuu.

Pääsynhallinta on osa tietokonejärjestelmän tietoturvaa. ATK-sanakirja määrittelee pääsynhallintaan kuuluvan ne ”toiminnot ja menettelyt, joiden avulla tietojärjestelmään pääsy tai tiedon saanti sallitaan vain valtuutetuille henkilöille tai sovelluksille” [Tie04]. Joka kerta kun käyttäjä kirjautuu sisään monikäyttöjärjestelmään, pääsynhallinta aktivoituu. Sen tarkoitus on rajoittaa mitä käyttäjä voi tehdä tietojärjestelmässä [SaS94]. Pääsynhallinnan tarkoitus on selitettävissä myös tarkastelemalla tietoturvariskejä. Tietoturvariskit voidaan jakaa karkeasti kolmeen eri kategoriaan: luottamuksellisuus (confidentiality), eheys (integrity) ja saatavuus (availability) [FKC07]:

- *Luottamuksellisuudella* tarkoitetaan sitä, että tietoa voivat käsitellä vain sellaiset henkilöt, joilla on siihen oikeus. Tähän kategoriaan kuuluvat mm. salasanat, salassa pidettävät kokouspöytäkirjat ja yrityksen sisäiseen käyttöön tarkoitetut taloustiedot.
- *Eheydellä* tarkoitetaan sitä, että tietoa ei pääse muuttamaan oikeudettomasti tai muutos pitää ainakin huomata.
- *Saatavuudella* tarkoitetaan sitä, että tieto on saatavilla silloin kun sitä tarvitaan.

Pääsynhallinta on elintärkeää informaation luottamuksellisuuden ja eheyden säilyttämiselle. Luottamuksellisuus edellyttää, että vain valtuutetut (authorized) käyttäjät pystyvät lukemaan tietoja, ja eheys sitä, että vain valtuutetut käyttäjät pystyvät kirjoittamaan tietoja luvallisella tavalla. Pääsynhallinnalla ei näyttäisi kuitenkaan olevan suoranaista yhteyttä saatavuuteen. On huomattavaa kuitenkin, että mikäli joku onnistuu murtautumaan luvatta järjestelmään, ei hänellä liene suuria vaikeuksia sammuttaa ko. järjestelmää. [FKC07]

2.1.2 Pääsynhallinnan peruskäsitteet

Pääsynhallinnan keskeisimmät termit ovat autentikointi eli todentaminen ja auktorisointi eli valtuuttaminen. Nämä termit aiheuttavat usein sekaannusta, joka johtuu niiden läheisestä suhteesta. Pääsyoikeuskäytännöllä (access control policy) tarkoitetaan niitä korkean tason ohjeita, joilla pääsyä valvotaan ja pääsyoikeuksista päätetään. Pääsyoikeusmekanismi (access control mechanism) on puolestaan joukko matalan tason ohjelmisto- ja laitteistofunktioita, joilla toteutetaan jokin pääsyoikeuskäytäntö. Pääsynhallintamalleja käytetään kuvaamaan pääsynhallintajärjestelmän tietoturvaominaisuuksia. Malli ei itsessään ota kantaa käytännön toteutukseen eikä ympäristöön, jossa sitä käytetään.

Tästä johtuen malleista ilmenevät yleensä vain tietoturvakonseptit ja pääsyoikeuskäytännöt, joita ne tukevat. [FKC07]

Lähes jokainen pääsynhallintamalli on kuvattavissa formaalisti käyttämällä käsitteitä käyttäjä, subjekti, objekti, operaatio ja käyttöoikeudet. Käyttäjällä tarkoitetaan tietojärjestelmän käyttäjää, joka on dialogissa järjestelmän kanssa istunnon välityksellä. Tietokoneessa suoritettavaan prosessiin, joka toimii käyttäjän puolesta, viitataan termillä subjekti. Käyttäjällä voi olla useita subjekteja käytössä yhtä aikaa, vaikka käytössä on vain yksi istunto. Jokainen käyttäjän käyttämä sovellus on subjekti ja niitä suoritetaan käyttäjän oikeuksilla edellyttäen, että käyttäjällä on niihin oikeus. [FKC07]

Objektilla tarkoitetaan tietokonejärjestelmän resursseja. Tällaisia ovat esimerkiksi tiedostot, hakemistot, tulostin ja tietokannat. Objektit ovat luonteeltaan passiivisia, jotka joko sisältävät tai vastaanottavat tietoa. Operaatio on puolestaan aktiivinen prosessi, jonka subjekti käynnistää. Varhaiset pääsynhallintamallit eivät tehneet eroa subjektin ja operaation välille, vaan kaikki aktiiviset prosessit olivat subjekteja. Roolipohjainen pääsynhallinta (Role Based Access Control) vaatii kuitenkin sen, että operaatiot ja subjektit ovat erillisiä. [FKC07]

Käyttöoikeudella tarkoitetaan valtuutusta suorittaa jokin ennalta määritelty toimenpide järjestelmässä. Käyttöoikeus ilmaistaan usein objektin ja operaation kombinaationa. Operaatio, joka suoritetaan kahdelle erilliselle objektille, vaatii kaksi erillistä käyttöoikeutta. Vastaavasti kaksi eri operaatiota, jotka suoritetaan yhdelle objektille, vaatii kaksi erillistä käyttöoikeutta. [FKC07]

2.1.3 Todentaminen, valtuuttaminen ja arviointi

Todentaminen on prosessi, jonka tarkoituksena on varmistaa, että käyttäjä on todella se, joka hän väittää olevansa. Tunnetuin ja eniten käytetyin todentamismenetelmä lienee salasanakysely. Todentaminen perustuu yhteen tai useampaan seuraavaan tekijään: [FKC07]

- Jotain mitä tiedät
- Jotain mitä omistat
- Jotain mitä olet

Salasanat, PIN-koodi ja lukkoyhdistelmät kuuluvat jotain mitä tiedät -kategoriaan. Yksinkertaiset todentamiset hoidetaan yleensä tällä menetelmällä, ja se on turvallinen niin pitkään kuin tunnistetieto pysyy salassa. Sähköiset henkilökortit, biopassit ja fyysiset avaimet ovat esimerkkejä jotain mitä omistat -kategoriasta. Jotain mitä olet -kategoriaan kuuluvat fyysiset ominaisuudet kuten esimerkiksi sormenjäljet, kasvojen piirteet ja silmien verkkokalvot.

Yksittäin käytettyinä edellä mainitut todentamismenetelmät eivät ole riittävän turvallisia. Esimerkiksi salasana voidaan arvata, avain kadottaa ja sormenjäljistä voidaan tehdä kopio. Tästä syystä erityistä turvallisuutta vaativat järjestelmät käyttävät useampaa todentamismenetelmää yhtä aikaa. Todentaminen on sitä vahvempaa mitä useampaa tapaa käytetään yhtä aikaa. Kahden (tai useamman) todentamismenetelmän yhtäaikaista käyttöä kutsutaan vahvaksi todennukseksi. Esimerkiksi onnistunut rahannosto pankkiautomaatilta edellyttää, että pankkikortti on saatavilla (jotain mitä omistat) ja että syötetty PIN-koodi (jotain mitä tiedät) on oikein. Todella arkaluontoiseen systeemiin pääsy voi vaatia kulkuluvan, ääninäytteen sekä kämmenjäljen. Tällainen erityisen hyvin suojattu järjestelmä on toteutettavissa jos suojauksen kohteena on esimerkiksi laitos tai yksittäinen ovi. [AlS04]

Todentamismenetelmien käyttöä täytyy aina arvioida kokonaisuutena ja käytännön vaatimusten mukaan. Hieman kärjistetysti voidaan sanoa, että ei ole järkevää käyttää verkkokalvoskannausta etuovella, jos sisään pääsee avonaisesta ikkunasta. Samasta syystä eri todentamismenetelmien yhdenmukainen käyttö on erityisen tärkeää. Epäjohdonmukaisuudet luovat tilaisuuksia tunkeutujille. Eräs tunnetuimmista epäjohdonmukaisista systeemeistä on kuvallinen luottokortti, vaikka siinä on käytössä kaikki neljä todentamismenetelmää: jotain mitä omistat (kortti), jotain mitä teet (allekirjoitus), jotain mitä tiedät (kortin numero, vanhentumispäivämäärä ja laskutusosoite) ja jotain mitä olet (kuva) [AlS04]. Neljän menetelmän käyttö vaikuttaa erittäin luotettavalta, mutta korttia pystyy kuitenkin käyttämään vain osalla näistä tiedoista. Verkkomaksamiseen riittää yleensä luottokortin numero, vanhentumispäivämäärä ja laskutusosoite. Joissain kaupoissa ostokset voidaan puolestaan maksaa pelkällä kortilla ja allekirjoituksella. Sähköi-

sen identiteetin hallinta ja identiteettivarkaudet rajataan tämän tutkielman ulkopuolelle. Aiheesta on kirjoitettu paljon ja yksi hyvä kirja aihepiiriin on Clare Sullivanin kirjoittama kirja Digital Identity [Sul11].

Valtuuttamisella rajoitetaan toimintoja, joita tietojärjestelmän laillinen käyttäjä voi suorittaa. Rajoituksen piirissä ovat käyttäjän itsensä suorittamat toiminnot kuin myös ohjelmat, joita suoritetaan kyseisen käyttäjän toimesta. Valtuuttamisella päätetään siis mihin ja minkälainen oikeus käyttäjällä on tietojärjestelmän resursseihin. On huomattavaa, että kunnollinen todentaminen on perusedellytys onnistuneelle valtuuttamiselle. Tämä johtuu siitä, että valtuuttamisen toimenpiteitä ei voida kohdistaa oikein jos käyttäjän identiteetti ei ole selvillä. [FKC07]

Pääsynhallinta ei ole kaikenkattava ratkaisu järjestelmän suojaukseen vaan sitä olisi perusteltua täydentää arvioinnilla eli auditoinnilla. Auditoinnilla tarkoitetaan käyttäjien kaikkien aktiviteettien ja pyyntöjen kirjaamista myöhempää tarkastelua varten. Tämä toimii osin pelotteena, koska käyttäjät tietävät, että heidän aktiviteettinsa kirjataan ylös, jolloin uskallus väärinkäytöksiin pienene. Samalla voidaan valvoa sitä, että valtuutetut käyttäjät eivät väärinkäytä oikeuksiaan. Auditoinnin avulla kirjattuja tapahtumia voidaan analysoida ja käyttää apuna väärinkäyttötapausten sekä niiden yrityksen etsimisessä ja järjestelmään jääneiden tietoturva-aukkojen paikantamisessa. Kuten valtuuttamisessakin, auditointi edellyttää toimivaa todentamista, jotta käyttäjän oikea identiteetti on selvillä. [SaS94]

2.1.4 Perusmallit

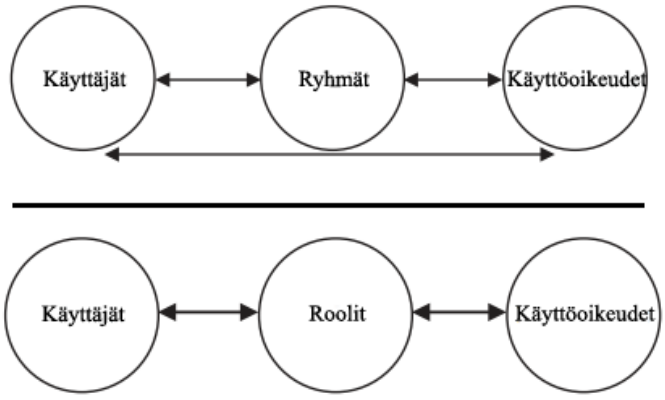
Harkinnanvarainen pääsynhallintamalli (discretionary access control, DAC) perustuu toimijoille myönnettäviin käyttöoikeuksiin ja lupiin. Tiedostojen tapauksessa tiedoston luoja on yleensä myös omistaja. Omistaja voi asettaa käyttölupia (esimerkiksi luku, kirjoitus) tiedostoon omien tarpeidensa mukaan, ja hän voi antaa myös näitä oikeuksia eteenpäin. Koska DAC mahdollistaa käyttöoikeuksien eteenpäin välityksen, se ei sovellu yksinään erityistä turvallisuutta vaativiin järjestelmiin. Järjestelmän turvallisuutta ei voida taata kaikissa oloissa, jos käyttäjät voivat antaa käyttöoikeuksia eteenpäin [HRU76]. DAC-mallin etuihin voidaan lukea joustavuus, koska oikeudet voidaan mää-

rittää varsin hienojakoisesti jokaiselle erikseen. Tästä on toisaalta haittaakin, koska käyttöoikeustietokanta voi kasvaa varsin isoksi ja sen hallinnointi voi käydä työlääksi.

DAC-mallin puutteita täydentämään kehitettiin pakollinen pääsynhallintamalli (mandatory access control, MAC). Siinä käyttöoikeuksia hallitaan järjestelmän tasolla eivätkä käyttäjät voi antaa käyttöoikeuksia eteenpäin. Koska käyttäjien toiminta on rajoitettua, MAC-malli takaa sen, että järjestelmä myös pysyy sellaisena. MAC-mallia sovelletaankin korkean tietoturvallisuuden järjestelmissä kuten sotilastietojärjestelmissä. [FKC07]

2.2 Roolipohjaisen pääsynhallinnan toimintaperiaate

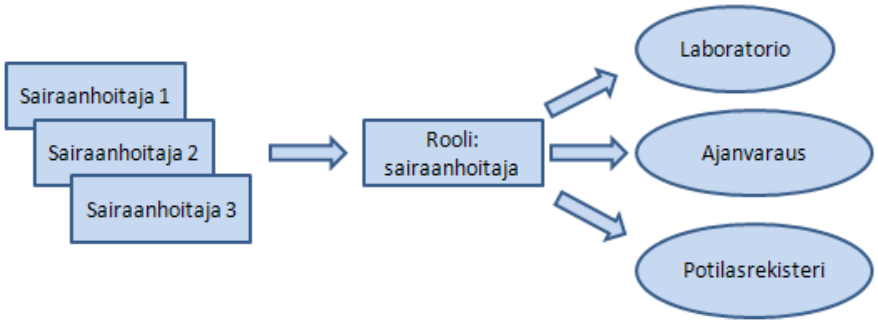
RBAC-mallissa käyttöoikeuksia ei anneta suoraan käyttäjille vaan käyttöoikeudet kuuluvat rooleille. Roolit ovat joukko käyttöoikeuksia, joita organisaation kuuluvat henkilöt tarvitsevat työnteossaan. Käyttäjien on mahdollista saada käyttöoikeudet vain roolien välityksellä, joihin heidät on sijoitettu (ks. kuva 1 alaosa). [FKC07]



Kuva 1: Roolien tuoma muutos käyttöoikeuksien sijoitukseen[mukaillen FKC07]

Esimerkiksi sairaalan kaikkien sairaanhoitajien voidaan ajatella kuuluvan sairaanhoitajarooliin. Sairaanhoitajarooli on valtuutettu käyttämään laboratoriojärjestelmää, ajanvarausjärjestelmää ja potilastietorekisteriä (ks. kuva 2). Kyseessä oleva järjestely mahdollistaa sen, että käyttöoikeudet täytyy määritellä vain kerran roolille sairaanhoitaja. Ilman rooleja jokaiselle yksittäiselle työntekijälle, joka toimii sairaanhoitajana, täytyisi antaa jokaisen käytettävän tietojärjestelmän käyttöoikeudet erikseen. Roolien käytöstä saavutetaan vastaava etu myös käyttöoikeuksia poistettaessa. Sairaanhoitajan vaihtaessa työpaikkaa riittää, että käyttäjätunnus poistetaan yhdestä paikasta (käyttäjän sairaanhoitaja-

rooli poistuu samalla). Suorilla käyttäjäoikeuksilla jokaisen eri tietojärjestelmän käyttöoikeudet jouduttaisiin poistamaan erikseen.



Kuva 2: Käyttöoikeuksien jakaminen roolien avulla.

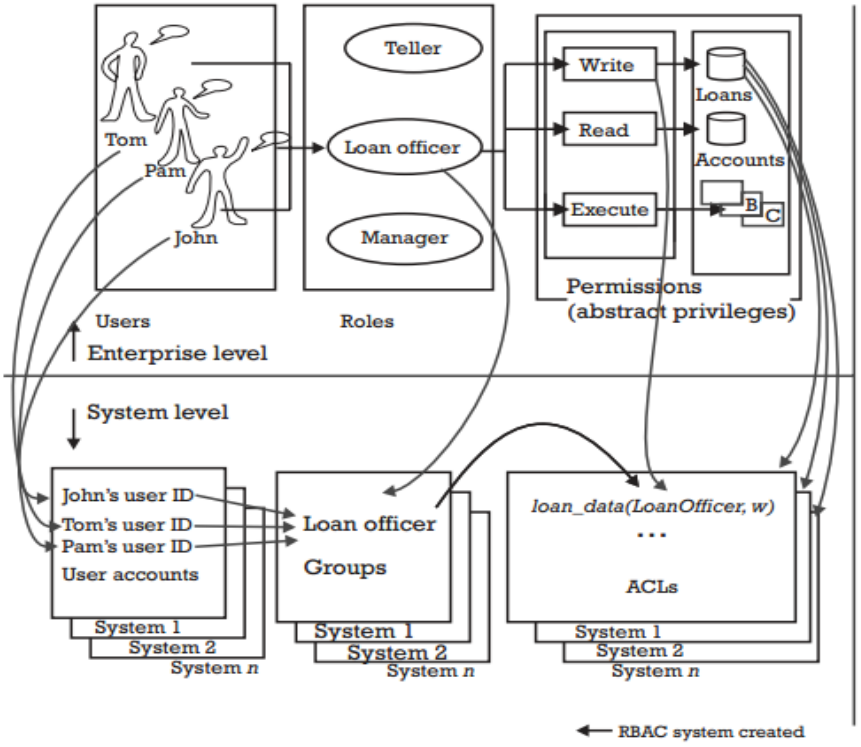
Roolipohjaisen pääsynhallinnan perusperiaate perustuu siihen huomioon, että organisaatioiden sisäiset roolit pysyvät suhteellisen vakioina. Käyttöoikeuksien ja käyttäjien on sitä vastoin huomattu muuttuvan paljon: esim. uusia ihmisiä rekrytoidaan ja uusia tietojärjestelmiä otetaan käyttöön. Näistä syistä johtuen roolien käyttäminen yksinkertaistaa käyttöoikeuksien hallinnointia ja katselmointia [FKC07]. Luvussa 2.7 tarkastellaan yksityiskohtaisemmin roolipohjaisen pääsynhallinnan tuomista eduista ja taloudellisista hyödyistä.

Yleisin tapa hallita käyttöoikeuksia on pääsyoikeuslistat eli Access Control Lists (*ACL*). Siinä kaikilla resursseilla on lista käyttäjistä, joilla on pääsyoikeus niihin. Tällaisia resursseja ovat esimerkiksi tiedostot, tulostimet ja hakemistot. Käyttäjät ovat yleensä yhdistetty ryhmiin, joita käytetään pääsyoikeuslistan yksittäisinä riveinä. Käyttöoikeuksia voidaan antaa sekä ryhmälle että käyttäjille (ks. kuva 1 yläosa). Tämä johtaa helposti tietoturvaongelmiin, sillä käyttöoikeuden poisto ryhmältä ei poista yksittäisen käyttäjän oikeutta kyseiseen resurssiin, jos hänellä oli se jo aiemmin. Roolipohjaisen pääsynhallinnan vaatimus siitä, että kaikki käyttöoikeudet tulevat roolien välityksellä poistaa tämän ongelman ja parantavaa tietoturvaa merkittävästi. [FKC07]

2.3 Yritystasolta järjestelmätasolle

Roolipohjainen pääsynhallinta eli Role Based Access Control (RBAC) toimii yritystasolla. Tästä johtuen jokaista yritystason käyttäjää varten RBAC-järjestelmä luo paikallisen käyttäjätunnuksen kuhunkin hallittavaan kohdejärjestelmään (ks. kuva 3). Vastaa-

vasti jokaista roolia varten luodaan paikallinen ryhmä kuhunkin kohdetietojärjestelmään. RBAC-järjestelmä hallinnoi näitä käyttäjätunnuksien ja ryhmien assosiaatioita ja tekee tarvittavat muutokset mikäli jotain muuttuu. Esimerkiksi yritystason käyttäjän poisto poistaa myös kaikki kyseistä käyttäjää vastaavat paikalliset käyttäjätunnukset ja ryhmäjäsenyyden eri järjestelmistä. [FKC07]



Kuva 3: Abstraktien operaatioiden ja resurssien systeemitason suhde [FKC07].

Järjestelmäriippumattomat käyttöoikeudet toteutetaan abstrakteilla operaatioilla abstrakteihin resursseihin. Näitä abstrakteja operaatioita ja resursseja vastaa yksi tai useampi natiivi operaatio ja resurssi järjestelmätasolla (ks. kuva 3). Natiivi operaatio voi olla esimerkiksi tiedoston luku, ja resurssia voi vastata esimerkiksi tietokanta. RBAC-järjestelmän tehtävänä on generoida näitä abstrakteja operaatioita ja resursseja vastaavat kohdejärjestelmän käyttöoikeudet käyttäen kohdejärjestelmässä käytössä olevia pääsynhallintamekanismeja. Kohdejärjestelmän pääsynhallinta voidaan toteuttaa monella eri tapaa eikä sen suinkaan tarvitse olla RBAC. Yleensä käytössä on jokin ACL-menetelmään perustuva pääsynhallintamekanismi. Kohdejärjestelmissä olevia natiiveja käyttöoikeuksia hallitaan kohdejärjestelmän hallintatyökaluilla. Näillä hallintatyökaluil-

la määritellään se mitä abstrakteilla operaatioilla ja resursseilla voidaan tehdä kohdejärjestelmässä. [FKC07]

2.4 Roolien luokittelutavat

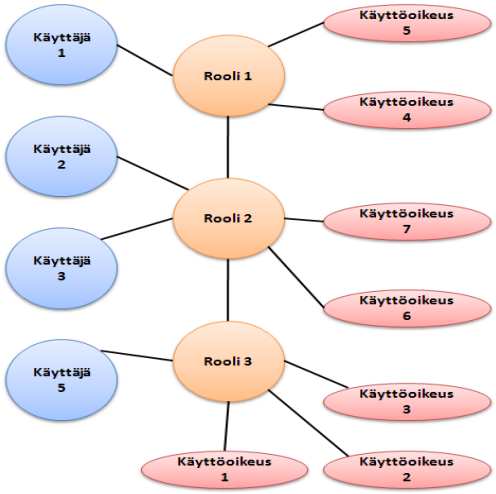
Roolit voidaan jakaa karkeasti kahteen eri luokkaan käyttötarkoituksen perusteella: järjestelmäroolit (system roles) ja toiminnalliset roolit (functional roles). Järjestelmäroolit ovat yksinkertaisia rooleja ja niitä käytetään järjestelmätason oikeuksia annettaessa. Ne muistuttavat paljon perinteisiä käyttöoikeusryhmiä. Järjestelmäroolien tehtävänä on koota yleisesti käytettyjä käyttöoikeuksia paremmin hallittaviin käyttöoikeuskokoelmiin. Tällaisia oikeuksia ovat esimerkiksi sovelluksen käynnistäminen tai luku- ja kirjoitusoikeus tiettyyn hakemistoon. Toiminnallisilla rooleilla tarkoitetaan puolestaan sitä mitä toimintoja roolilla pystytään tekemään ohjelman sisällä. Ne perustuvat yleensä organisaatiorakenteeseen kuten käyttäjän työtehtäviin, asemaan tai sijaintiin organisaatiossa. Toiminnalliset roolit voivat koostua toisista toiminnallisista rooleista ja järjestelmärooleista. Esimiehellä voi olla esimerkiksi tarvittava toiminnallinen rooli matkalaskujen hyväksyntään, joka puuttuu tavallisilta työntekijöiltä. On huomattavaa, että käyttäjällä voi olla useita järjestelmärooleja ja toiminnallisia rooleja. Joissain tapauksista eri roolien yhteisvaikutus voi johtaa ei-toivottuun pääsyoikeuskombinaatioon, jonka eri estämistapoja tarkastellaan lähemmin luvussa 2.6. [FKC07]

Edellä esitetty jako toiminnallisiin rooleihin ja järjestelmärooleihin ei ole ainoa tapa kategorisoida rooleja. Erityisesti suurissa organisaatioissa roolien hallinta voi käydä hankalaksi ja hallinnointia helpottamamaan on kehitetty komposiittimalli (composite model) [PCN04]. Komposiittimallissa roolit jaetaan kolmeen eri luokkaan: organisaatioroolit, yritysroolit ja tietojärjestelmäroolit. Mallin ideana on erottaa systeemitason roolit organisatorista rooleista ja tarjota näiden välille yhtymäkohta. Vastaavasti Roeckle ja kumppanit [RSW00] esittävät hieman komposiittimallista poikkeavan jaottelun, jossa roolit jaetaan viiteen eri luokkaan: funktionaaliseen, erikois-, organisaatio- ja perusrooleihin sekä hierarkkisiin rooleihin.

2.5 Roolihierarkiat

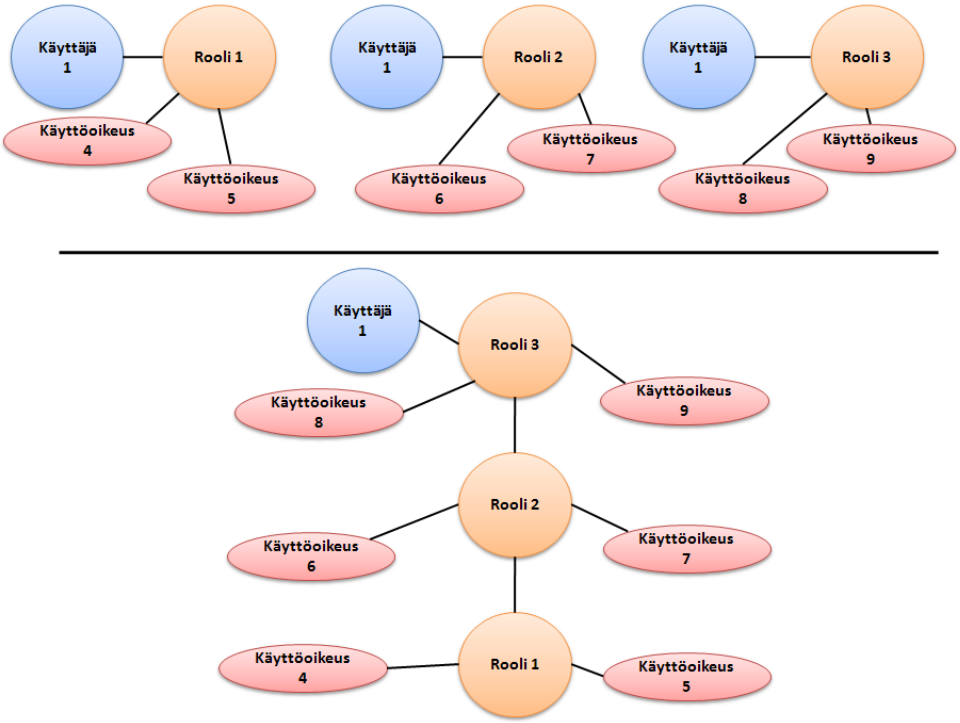
Roolit voidaan määritellä erillään toisista, mutta yleisemmin käytetään roolihierarkioita. Roolihierarkioiden käyttö perustuu siihen havaintoon, että usein täysin erillisillä rooleilla on paljon yhtenäisiä käyttöoikeuksia. Äärimmäisessä tapauksessa jokin tietty käyttöoikeus voi kuulua kaikille organisaation työntekijöille. Olisi erittäin tehotonta ja epäkäytännöllistä, jos tällaisen käyttöoikeuden joutuisi lisäämään erikseen kaikkiin käytössä oleviin rooleihin. Vastaavasti työmäärä pienenee, kun käyttäjiä ei tarvitse lisätä useaan yleiskäyttöiseen roolin. Roolihierarkioita käyttämällä on mahdollista koota työtehtävän suorittamisessa tarvittavat käyttöoikeudet perimällä niitä alempana roolihierarkiassa olevilta rooleilta. Näin syntynyttä roolia voidaan uudelleenkäyttää muodostettaessa uusia rooleja, joiden osana nämä samat käyttöoikeudet ovat. [FKC07]

Roolihierarkioista puhuttaessa käytetään yleensä seniori- ja junioriroolin käsitettä [CoD07]. Seniorirooli on korkeammalla hierarkiassa kuin junior irooli. Kaavioissa tämä ilmenee yleensä siten, että senioriroolit piirretään junioriluokan yläpuolelle. Periytymistä tapahtuu kahteen suuntaan: rooliin jäsenyys periytyy alaspäin ja vastaavasti käyttöoikeudet periytyvät ylöspäin. Tällä tarkoitetaan sitä, että senioriroolit perivät kaikki junioriroolien käyttöoikeudet riippumatta siitä kuinka alhaalla hierarkiassa junior irooli sijaitsee. Vastaavasti junioriroolit perivät kaikki ne käyttäjät, jotka kuuluvat sen yläpuolella olevaan senioriroolliin. Kuvassa 4 roolilla 1 on käyttöoikeudet 1, 2, 3, 6, 7, 4 ja 5. Rooliin 3 kuuluu käyttäjät 1, 2, 3 ja 5.



Kuva 4: Käyttöoikeuksien ja roolijäsenyyden periytyminen roolihierarkiassa [mukaillen CoD07].

Kuvasta 5 käy ilmi roolihierarkian tuoma tehokkuus käyttäjien sijoituksesta rooleihin. Kuvan yläosassa on kolme erillistä ei-hierarkkista roolia, joihin on sijoitettu sama käyttäjä. Roolien käyttöoikeuksista nähdään, että käyttäjä saa käyttöoikeudet 4-9. Kuvan alaosassa on sama tilanne käyttäen hyväksi roolihierarkioita. Alaosan hierarkioita käyttävässä tapauksessa selvitään yhdellä käyttäjän sijoituksella.



Kuva 5: Erillisten roolien mallintaminen roolihierarkian avulla [mukaillen CoD07].

Roolihierarkiat tuovat mukanaan tiettyä hallinnan mukavuutta, mutta ne eivät ole välttämättömiä. Roolihierarkioiden käyttöä tulee harkita silloin kun monet käyttäjät tarvitsevat samankaltaisia käyttöoikeuksia. Näistä yhteisistä käyttöoikeuksista voidaan muodostaa junioririrooleja, joita voidaan sijoittaa tarvittaviin kohtiin roolihierarkiassa. Näin vältetään sijoittamasta samanlaisia käyttöoikeuksia moneen rooliin. Coynen mukaan roolihierarkioita ei tulisi käyttää silloin kun käytettäviä rooleja on suhteellisen vähän. Toisaalta senioriroolin tulisi välittää ainakin 40 prosenttia käyttöoikeuksista junioriroolleille, jotta roolihierarkian tuoma kompleksisuus olisi hyväksyttävissä. [CoD07]

Roolihierarkioiden käyttö voi tuoda mukanaan myös potentiaalisia ongelmia. Atomisten eli hyvin vähän käyttöoikeuksia sisältävien roolien käyttäminen hierarkioita muodostet-

taessa voi vaikuttaa järkeenkäyvältä. Liian atomisilla rooleilla ei kuitenkaan ole vastinetta reaalimaailmassa. Näin ollen niihin tuskin tullaan lisäämään käyttäjiä tai ainakin käyttäjille sopivien roolien etsiminen on vaikeaa. Tämä sotii roolipohjaisen pääsynhallinnan käyttäjähallinnan helppoutta vastaan. Toinen potentiaalinen ongelma on se, että roolien periytymissuhde voi muodostua monimutkaiseksi ja vaikeasti ymmärrettäväksi. Tietoturvan kannalta yksinkertaisuus on hyvästä. [CoD07]

Mikäli roolihierarkioiden kompleksisuus nousee ei-hyväksyttävälle tasolle, voidaan hierarkioista luopua. Eräs tapa on luoda rooli jokaiselle työtehtävälle erikseen ja liittää rooleihin niissä tarvittavat käyttöoikeudet. Roolien määrä kasvaa huomattavasti, mutta roolien hallinnointi voidaan jakaa usean ylläpitäjän kesken. Hallinnointikustannukset ovat kuitenkin edelleen paljon pienempiä kuin suoria käyttöoikeuksia käytettäessä. Jokaiselle käyttäjälle voidaan luoda myös oma rooli, jolloin ei muodostu hierarkioita. Tällöin kuitenkin menetetään roolinpohjaisen pääsynhallinnan tuoma hallinnollinen etu, joka on yksi roolipohjaisen pääsynhallinnan kulmakivistä, joten se ei ole järkeenkäypää. [CoD07] Luvussa 2.7 perehdytään tarkemmin roolipohjaisen pääsynhallinnan tuomiin kustannushyötyihin ja etuihin.

2.6 Mallit ja standardit

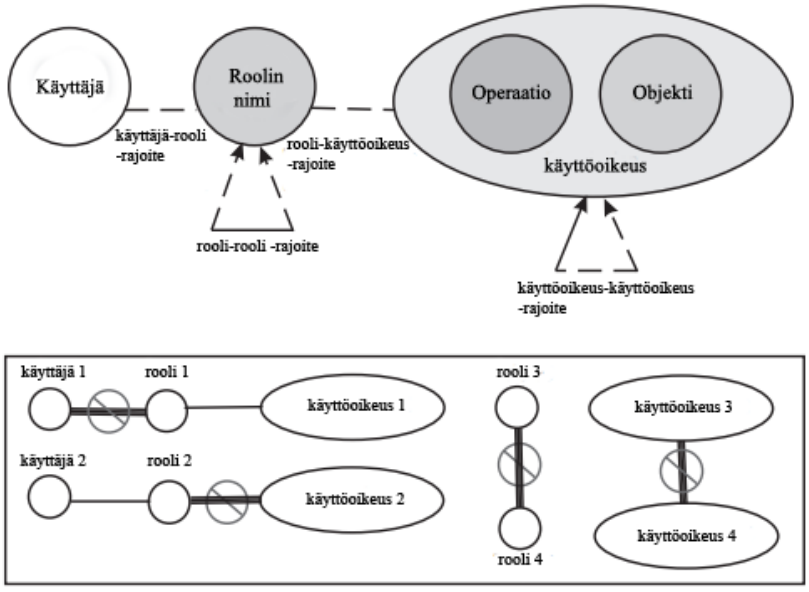
Roolipohjaisen pääsynhallinnan tuomien etujen maksimoimiseksi tarvitaan yhteisiä pelisääntöjä sen toiminnasta. Tässä luvussa luodaan kronologinen katsaus eri RBAC-malleihin alkaen Ferraiolo ja Kuhnin vuonna 1992 julkaisemasta mallista, jota pidetään ensimmäisenä yhdistettynä RBAC-mallina, päätyen vuonna 2004 julkistettuun standardiin. Luvussa sivutaan myös kirjoitushetkellä vireillä olevia luonnoksia, jotka eivät kuitenkaan ole vielä saavuttanut standardin asemaa.

2.6.1 Ferraiolon ja Kuhnin malli

Ensimmäisen RBAC-mallin esittivät Ferraiolo ja Kuhn vuonna 1992 [FeK92]. Mallin tarkoitus oli korjata MAC- ja DAC-malleissa ilmenneitä ongelmia ja se oli tarkoitettu ensisijaisesti yritysmaailmaan tarpeisiin. Mallilla pyrittiin helpottamaan käyttöoikeuksien ja käyttäjien hallinnointia sekä parantamaan tietoturvaa. Kiinnostuksen kohteena ei

ollut enää yksittäiset tiedostot tai muut resurssit, joihin käyttäjillä oli pääsyoikeus, vaan se, mitä operaatioita käyttäjät pystyisivät näillä resursseilla tekemään. [FeK92]

Ferraiolon ja Kuhnin mallissa otettiin käyttöön tietoturvallisuudessa käytetyt kaksi keskeistä periaatetta: vähäisimpien oikeuksien periaate (Least Privilege) ja vastuiden eriyttäminen periaate (Separation of Duties). Ensimmäisellä tarkoitetaan sitä, että työntekijällä pitäisi olla vain ja ainoastaan ne käyttöoikeudet, joita hän tarvitsee suorittaessaan työtehtäviä. Yleensä käyttöoikeuksia on annettu ”varmuuden varalle”, ja ajan mittaan tällaiset ylimääräiset käyttöoikeudet voivat aiheuttaa kumuloituessaan ei-toivottuja käyttöoikeuskombinaatioita. Jälkimmäinen periaate pyrkii välttämään tilanteita, joissa työntekijöillä on mahdollisuus vilpillisiin toimiin hallussa olevilla käyttöoikeuksilla. Esimerkiksi laskujen kirjoitus ja niiden hyväksyntä eivät yleensä ole saman henkilön vastuulla. [FeK92]



Kuva 6: Rajoitteiden esiintymismahdollisuudet [mukaillen CoD07].

Roolipohjaisessa pääsynhallinnassa vastuiden eriyttämistä voidaan toteuttaa kahdella eri tavalla. Staattisessa eriyttämisessä kielletyt yhdistelmät asetetaan kiinteästi rooleja ja käyttöoikeuksia määriteltäessä. Dynaamisessa eriyttämisessä kielletyt yhdistelmät tarkastetaan suorituksen aikana. Rajoitteita voidaan asettaa neljään eri kohtaan: roolien välille, käyttäjän ja roolin välille, roolin ja käyttöoikeuden välille ja käyttöoikeuksien välille (ks. kuva 6). [CoD07]

2.6.2 Referenssimallit

Sandhu ja kumppanit [SCF96] esittivät neljä referenssimallia kuvaamaan roolipohjaisen pääsynhallinnan eri osa-alueita vuonna 1996 (ks. taulukko 1).

Taulukko 1: Referenssimallien keskeiset ominaisuudet

Perustaso (**$RBAC_0$**)	käyttäjät, roolit, käyttöoikeudet ja istunnot
Roolihierarkiat (**$RBAC_1$**)	seniorirooli, juniorirooli
Rajoitteet (**$RBAC_2$**)	vähäisimpien oikeuksien ja vastuiden eriyttämisen periaate
Yhdistetty malli (**$RBAC_3$**)	kaikki edellä mainitut koottuna

Perustaso ($RBAC_0$) vastaa hyvin pitkälti Ferraiolon ja Kuhnin mallia ja se kuvaa peruskäsitteet kuten: käyttäjät, roolit, käyttöoikeudet ja istunnot (sessions). Istunnolla tarkoitetaan käyttäjällä aktiivisena olevien roolien joukkoa. $RBAC_1$-taso lisää perustasoon mahdollisuuden roolihierarkioiden käyttöön. Rajoitteet ($RBAC_2$) mahdollistavat puolestaan vaarallisten yhdistelmien estämisen. Yhdistetty malli ($RBAC_3$) kokoaa aikaisemmat mallit ($RBAC_0$, $RBAC_1$, $RBAC_2$) yhdeksi yhdistetyksi malliksi. [SCF96]

2.6.3 Mallista standardiksi

Eri toimijat olivat alkaneet käyttämään referenssimalleissa esitettyjä käytäntöjä, mutta käytöstä puuttui yhteinen standardi, joka määrittelisi yhdenmukaisesti ja järjestelmällisesti roolipohjaisen pääsynhallinnan ominaisuudet. Laajasti hyväksytyn mallin puute aiheutti epävarmuutta ja hämmennystä roolipohjaisen pääsynhallinnan tuomista hyödyistä ja tarkoituksesta. Näistä syistä johtuen NIST (National Institute of Standards and Technology) alkoi kehittää RBAC-standardia. [FKC07]

Ensimmäinen luonnos julkaistiin vuonna 2000 ja toinen versio vuonna 2001. INCITS (The InterNational Committee for Information Technology Standard) myönsi NIST:in aloittamalle työlle standardin aseman vuonna 2004. Role Based Access Control-standardi (ANSI INCITS 359-2004) oli syntynyt. Standardi perustui edellä esitettyihin

referenssimalleihin. Merkittävänä lisänä standardissa kuvattiin myös toiminnalliset kuvaukset, joita käytetään suunniteltaessa roolipohjaista pääsynhallintaa tukevia sovelluksia. [Ans04]

RBAC-standardin kehitystyö jatkuu edelleen. INCITS julkaisi vuonna 2008 RIIS-luonnoksen (An RBAC Implementation and Interoperability Standard), joka pyrkii täydentämään aiemmin esitettyä standardia. Sen tavoitteena on kuvata tarkemmalla tasolla kuinka ohjelmistoista tehdään RBAC-standardin mukaisia. Standardi pyrkii myös helpottamaan vertailua eri RBAC-tuotteiden kesken, sekä mahdollistaa tuotteiden yhteistoiminnan. Standardi on tutkielman kirjoitusvaiheessa vielä luonnosvaiheessa. [CoW08]

RBAC-standardiin on esitetty lukuisia muitakin laajennuksia eri käyttötarpeita varten, mutta ainakaan tutkielman kirjoitushetkellä yhtäkään niistä ei ole hyväksytty mukaan. Tästä syystä kyseisiä laajennuksia ei käsitellä tässä tutkielmassa tarkemmin.

2.7 Roolipohjaisuuden hyödyt

Roolipohjainen pääsynhallinta mahdollistaa keskitetyn käyttäjän- ja pääsynhallinnan. Monissa organisaatioissa ja yrityksissä loppukäyttäjät eivät omista informaatiota, joihin heille on annettu käyttöoikeus, vaan informaatio on luonteeltaan organisaation omaisuutta. Perinteinen harkinnanvarainen pääsynhallintamalli ei tällöin ole välttämättä sopivin vaihtoehto. [Vir96]

RBAC poikkeaa perinteisistä pääsynhallintamalleista toimimalla korkeammalla abstraktiotasolla. Pääsynhallinta ei perustu yksittäisten dataobjektien kontrollointiin vaan rooleihin, joita hallinnoidaan tasolla, joka vastaa organisaatiorakennetta. Tämä abstrahointi tuo mukanaan hallinnollisia etuja ja mahdollisuuksia, joita ei ole perinteisissä menetelmissä. Esimerkiksi uuden käyttäjän käyttöoikeuksien asettaminen hoidetaan sijoittamalla hänet yksinkertaisesti vain ennaltamääriteltyihin rooleihin. Tämä toimenpide ei vaadi it-alan ammattilaista vaan onnistuu hyvin esimerkiksi esimieheltä tai henkilöstöhallinnolta. Uudelle työntekijälle tehokkuus näkyy siinä, että tarvittavat tunnukset ja käyttöoikeudet ovat heti käytettävissä eikä usein tapahtuvaa käyttöoikeuksien odottelua esiinny. [Vir96]

Roolipohjaisen pääsynhallinnan on tutkitusti todettu vähentävän tietoturvan hoitamisesta aiheutuvia kustannuksia ja työmääriä. Ero on merkittävä etenkin suurissa organisaatioissa ja pankeissa. Näissä organisaatioissa on tyypillistä, että tietoturvan ylläpitämisen aiheuttavat kustannukset ylittävät merkittävästi tietomurroista aiheutuvat kustannukset. Tämän johdosta mikä tahansa pääsynhallintamalli tai työkalu, joka yksinkertaistaa tietoturvan hallinnointia, ja näin ollen parantaa sen tuottavuutta, on tavoiteltavaa. Roolipohjaisuuden on myös itsessään todettu kohentavan tietoturvaa. Vastuiden eriyttämisen ja vähäisimpien oikeuksien tietoturvaperiaatteet ovat nimenomaan tietoturvan parantamiseen tähtääviä konstruktioita. [Vir96]

On kuitenkin huomattavaa, että roolipohjaisella pääsynhallinnalla on myös omat rajoitteensa. RBAC ei sinällään ota kantaa esimerkiksi siihen kuinka roolit tulisi määritellä, kuinka käyttöoikeudet tulisi jakaa tai missä tilanteissa rajoitteita tulisi käyttää. Nämä kuuluvat roolien määrittelyn piiriin, joka on luonteeltaan luova prosessi. Tästä johtuen siinä voi tapahtua myös virheitä, jotka vaikuttavat epäsuotuisasti lopputulokseen. [SCF96] Luvuissa kolme ja neljä esitetään tekemäni systemaattinen kirjallisuuskatsaus, jossa arvioidaan tieteellisessä kirjallisuudessa esiintyviä tapoja roolien määrittelyn suorittamiseen ja vertaillaan niiden käyttökelpoisuutta.

Kaikkia tietojärjestelmissä olevia objekteja ei ole tarkoituksenmukaista hallinnoida pelkästään roolipohjaisella pääsynhallinnalla (ks. kuva 7). Monessa tapauksessa parhaaseen lopputulokseen pääsee yhdistämällä eri pääsynhallintamalleja.

Kuva 7: Tietoturvapolitiikka, joka on toteutettu useilla pääsynhallintamalleilla [mukaillen CoD07].

Esimerkiksi paikalliset tulostimet, kotihakemistot yms. voidaan hallinnoida käyttöjärjestelmän omilla pääsynhallintamekanismeilla. Vastaavasti taas yrityksen kattava tuntikirjaus- ja laskutusjärjestelmä ovat esimerkkejä tilanteista, joihin roolipohjaisuus sopii hyvin. [CoD07] Kuvan 7 tietoturvapolitiikka on hieman kärjistetty, ja ainakaan aivan isoimpia yrityksiä ja organisaatioita lukuun ottamatta, käytössä tuskin on aivan näin monta pääsynhallintatapaa.

3 ROOLIEN MÄÄRITTELYMENETELMÄT

Tässä luvussa esitetään tekemääni systemaattista kirjallisuuskatsausta roolien määrittelystä. Katsauksen tarkoituksena oli kartoittaa tieteellisessä kirjallisuudessa esiintyvät menetelmät suorittaa roolien määrittelyprosessi ja arvioida niiden käyttökelpoisuutta. Luvussa käydään läpi aineiston hankintaprosessi, käytetyt haut sekä artikkeleille asetetut kriteerit. Varsinaiset tulokset esitetään luvussa neljä.

3.1 Aineiston hankinta

Aineisto haettiin systemaattista kirjallisuuskatsausta varten käyttäen hyväksi Itä-Suomen yliopiston tarjoamaa Nelli-portaalia. Nelli-portaali on tiedonhakujärjestelmä, joka sisältää Itä-Suomen yliopiston ja Kuopion yliopistollisen sairaalan käytettävissä olevat sähköiset aineistot. Näitä ovat mm. eri viitetietokannat ja sähköiset lehdet. Valitsin käyttämäni tietokannat aikaisemmilla opintojaksoilla tutuiksi tulleista tunnetuista tietokannoista, joihin minulla oli pääsy. Haun kattavuutta pyrin lisäämään käyttäen hyväksi Google Scholar -hakumoottoria ja suorittamalla sillä alustavia hakuja aihepiirin tiimoilta. Näin löydetyistä artikkeleista poimin käytetyn tietokannan mukaan varsinaiseen hakuun. Valitsemani kansainväliset tietokannat olivat ACM, IEEE Xplore, ScienceDirect ja Web of Science. Haut suoritin jokaiselle valitsemalleni tietokannalle erikseen käyttäen Nelli-portaalia. Nelli-portaalin hakuparametrien syöttö oli jokseenkin rajoittunut, joten kokeilin hakuja myös tietokantojen omilta verkkosivuilta. En havainnut tutkimukseni kannalta eriäviä hakutuloksia, joten toistettavuuden helpottamiseksi suoritin lopulliset haut Nelli-portaalilla. Näin menettelemällä käyttämäni haut voidaan suorittaa helposti uudelleen yhdessä paikassa.

Aineiston sisäänottokriteereinä oli se, että artikkelin tuli olla tieteellinen artikkeli ja sähköisesti saatavilla. Aineiston hyväksymiskriteerit olivat seuraavat:

1. Tutkimuskohteena on *roolien määrittely tai jokin apukeino roolien määrittelyyn*.
2. Tutkimuksen *julkaisuajankohta on vuonna 1992 tai sen jälkeen* julkaistut artikkelit. Kuhnin ensimmäinen yhdistetty sovellusriippumaton malli julkaistiin vuonna 1992. Aikaisemmat RBAC-mallit ovat sovelluskohtaisia.

3. Tutkimus on *julkaistu tunnetussa tieteellisessä julkaisussa englannin kielellä*.

Poissulkukriteereinä olivat seuraavat:

1. Tutkimuksen pääsisältö on *formaali algoritmi roolien määrittelyyn tai käyttöoikeuksien etsintään*. Formaalit algoritmit ovat rajattu tutkimuksen ulkopuolelle, koska ne keskittyvät vain hyvin kapeaan roolien määrittelyn osa-alueeseen. Jos tutkimuksessa esitetään muutakin kuin pelkkä algoritmi, voidaan se hyväksyä, mikäli muut kriteerit täyttyvät.
2. Tutkimuskohteena on *laajennos tunnettuihin RBAC-malleihin tai -standardiin*, joka ei ole vakiinnuttanut asemaansa.
3. Tutkimuskohteena on *tutkimuksessa julkaistu sovellus, tai tutkimus nojaa pelkästään kaupalliseen sovelluksen toiminnan esittelyyn*.

Roolipohjaista pääsynhallintaa on tutkittu laajasti, ja näin ollen myös julkaistuja artikkeleita löytyi paljon. Erilaisia RBAC-standardin laajennoksia oli myös lukuisasti. Roolien määrittelystä on myös julkaistu lukuisia artikkeleita, mutta suurin osa niistä käsittelee formaaleja algoritmeja, jotka ovat rajattu tutkimuksen ulkopuolelle. Edellä mainituista syistä johtuen tutkimusongelmiini vastauksia antavien artikkelien seulominen massasta oli paikoin haastavaa.

Kokeilin erilaisia hakusanoja ja niiden eri kombinaatioita jokaisessa käyttämässäni tietokannassa. Kokeilemiani hakusanoja olivat mm. *rbac, role, role engineering, role management, role administration ja role engineering process* sekä näiden erilaiset yhdistelmät. Aloitin hakujen tekemisen ensin yksittäisillä sanoilla saadakseni jonkinlaisen käsityksen aihepiiristä kirjoitetuista artikkeleista. Mielenkiintoisten artikkelien esiinsaaminen vaati kuitenkin useamman hakusanan käyttöä, jotta hakutulosten läpikäymiseen ei olisi mennyt kohtuuttomasti aikaa tutkimuksen laajuuteen nähden.

Käyttämäni tietokannat, hakusanat ja hakutulokset olen kerännyt liitteisiin (ks. liite 1). ACM-tietokannan osalta olen sisällyttänyt hakutuloksiin myös alustavia hakuja, joita kokeilin ja kävin otsikkotasolla läpi. Hakutulosjoukko kyseisillä hakusanoilla oli kuitenkin liian laaja systemaattiseen läpikäymiseen, jonka johdosta kyseisten hakujen kohdalla ei ole hyväksyttyjen artikkeleiden määrää. Liite 1 sisältää myös yhden hakutuloksen Google Scholar -hakumoottorilla. Kyseiseen artikkeliin oli viittauksia sisäänotetuis-

sa artikkeleissa, mutta CiteSeerX-nimistä digitaalista kirjastoa ei pysty käyttämään Nellin kautta eikä artikkelia löytynyt käyttämistäni tietokannoista. Tästä johtuen kyseinen artikkeli on listattuna erillisessä taulukossa, jossa mainitaan artikkelin nimi ja julkaisufoorumi. CiteSeerX-kirjaston verkko-osoite on *http://citeseerx.ist.psu.edu/index*.

Haku ACM-tietokannasta tuotti 477 osumaa, jotka kävin ensin otsikkotasolla läpi. Otin mukaan neljä artikkelia suoraan otsikon perusteella. Viitteistä poimin mukaan kaksi artikkelia. ACM-tietokannasta valittujen artikkeleiden määrän suuri suhteellinen osuus on perusteltavissa sillä, että pääsynhallinnan tieteellisistä tuloksista raportoiva konferenssi SACMAT (Symposium on Access Control Models and Technologies) kuuluu ACM-järjestöön.

Haku IEEE Xplore -tietokannasta tuotti 39 osumaa, jotka kävin otsikko- ja tiivistelmätasolla läpi. Hyväksymiskriteereitä täyttäviä artikkeleita ei löytynyt suoraan, mutta muiden artikkeleiden viitteiden perusteella valitsin mukaan yhden. ScienceDirect-tietokanta tuotti 270 osumaa, jotka kävin otsikkotasolla läpi. Mukaan valikoitui yksi mielenkiintoinen artikkeli, jonka rajasin kuitenkin lukemisen jälkeen pois. Web of Science -tietokanta tuotti 17 osumaa, joiden tiivistelmän luin läpi. Yksi aiemmin mukaanotettu artikkeli löytyi, mutta uusia osumia ei tullut. Edellä mainittujen tietokantojen lisäksi etsin ja luin Google Scholar -hakumoottorilla yhden sisäanotetuissa artikkeleissa mainitun artikkelin. Kyseinen artikkeli täytti sisäänottokriteerit..

Systemaattisella kirjallisuushaulla sain tutkimuksen aineistoksi yhteensä 8 tieteellistä artikkelia, jotka täyttivät hyväksymiskriteerit. Käytetyt artikkelit ovat listattuna liitteissä (ks. liite 2). Artikkeleiden määrän vähäisyys toi esille mahdollisen ongelman aineiston kattavuudessa, mutta vähäisyys johtunee todennäköisesti vain tutkimuksen puutteesta [RSW00; NeS02].

3.2 Aineiston analysointi

Aineiston analysointi tapahtui maaliskuussa 2012. Tarkoituksenani oli suorittaa koko analysointiprosessi täysin sähköisesti tulostamatta yhtään artikkelia paperille. Aineiston haut tein kotona pöytätietokoneella, mutta varsinaisen artikkeleiden luku tapahtui iPad-taulutietokoneella hyödyntäen GoodReader-nimistä lukusovellusta. Kyseinen sovellus

mahdollistaa mm. omien muistiinpanojen, alleviivauksien ja korostuksien lisäämisen käsiteltäviin dokumentteihin. Taulutietokoneen käytön eduksi voidaan laskea myös lukupaikan vapaampi valinta.

Ensimmäisellä lukukerralla luin artikkelit läpi saadakseni yleiskuvan aineistosta. Tämän jälkeen aloin miettimään tutkimusongelmiani, ja sitä mitä uutta kukin artikkeli toi roolien määrittelyyn, ja mihin roolien määrittelyn vaiheeseen se liittyi. Näihin kysymyksiin vastaaminen edellytti useita lukukertoja, joiden aikana tein paljon muistiinpanoja ja alleviivasin artikkeleiden ydinkohtia.

Artikkelit lähestyivät roolien määrittelyä hyvin erilaisista lähtökohdista (ks. taulukko 2). Osassa artikkeleissa keskityttiin pelkästään johonkin pieneen yksityiskohtaan kuten havainnollistamistapoihin, kun toisissa esitettiin kokonaisia prosesseja onnistuneeseen roolien määrittelyn suorittamiseen. Kaikkein laajimmissa prosesseissa yhdisteltiin useita erilaisia lähestymistapoja, ja apuna käytettiin myös tähän tarkoitukseen kehitettyjä ohjelmistoja.

Taulukko 2: Roolien määrittelyn lähestymistavat

Tekijä	**Kuvaus**
Coyne	9 vaiheen yksinkertainen menetelmä roolien, käyttöoikeuksien, rajoitteiden ja hierarkioiden määritykseen
Fernandez & Hawkins	Käyttötapauksien (Use Cases) hyödyntäminen roolien määrittelyssä.
Thompsen ja kumpp.	Komponenttiteknologiaa hyödyntävä malli
Roeckle ja kumpp.	Prosessilähtöinen roolien määrittelyprosessi
Kern ja kumpp.	Roolin elinkaareen perustuva malli
Strembeck ja kumpp.	Skenaariolähtöinen roolien määrittelyprosessi
He ja kumpp.	Tavoitelähtöinen roolien määrittelyprosessi
Giblin ja kumpp.	Integroitu roolien määrittelyprosessi

Kronologisesti tarkasteltuna roolien määrittelyn keinot ovat muuttuneet vuosien saatossa moninaisemmiksi ja ne ammentavat hyväksi havaittuja keinoja muualta ohjelmistotuotannon kentältä kuten vaatimusmäärittelystä ja elinkaarimallista. Luvussa neljä tutkitaan mitä nämä eri menetelmät pitävät sisällään, ja miten ne helpottavat roolien määrittelytyötä.

4 ROOLIEN MÄÄRITTELYMENETELMINEN VERTAILU

Tässä luvussa esittelen suorittamani systemaattisen kirjallisuuskatsauksen tulokset. Jokainen roolien määrittelymenetelmä kuvataan periaatetasolla ja niiden vahvuuksia ja heikkouksia pyritään tuomaan esiin. Tämän jälkeen eri menetelmiä vertaillaan keskenään käyttäen apuna kuutta vertailukriteereitä. Luvun lopuksi pohditaan saatuja tuloksia ja muodostetaan yhteenveto tutkittujen menetelmien sopivuudesta roolien määrittelyyn.

4.1 Tutkimusaineiston julkaisuvuodet

Systemaattinen kirjallisuuskatsaus tuotti aineistoksi 8 tieteellistä artikkelia. Vanhin artikkeli oli vuodelta 1995 ja uusin vuodelta 2010. Artikkelien julkaisuvuodet ovat taulukossa 3.

Taulukko 3: Artikkeleiden julkaisuvuodet

Vuosi	95	96	97	98	99	00	01	02	03	04	05	06	07	08	09	10	11
Artikkelit (kpl)	1	0	1	1	0	1	0	2	1	0	0	0	0	0	0	1	0

Artikkeleiden julkaisuvuodet eivät ole kovin tasaisesti jakautuneet katsauksessa käytetyn aikarajauksen mukaan (1992–2012). Mielestäni vuosina 1992–1995 keskityttiin vielä roolipohjaisen pääsynhallinnan teknisiin yksityiskohtiin ja olemukseen, ja vasta vuonna 1997 esitetty Sandhun ja kumpp. referenssimallit [SCF96] loivat tarvittavan pohjan roolien määrittelyn problematiikan tutkimiseen.

4.2 Roolien määrittelyn menetelmät

Tässä luvussa esitellään kirjallisuuskatsauksessa esiintulleet menetelmät, joiden tarkoituksena on helpottaa roolien määrittelytyötä. Tämän jälkeen tutkittuja menetelmiä vertaillaan keskenään, tehdään arvioita menetelmien sopivuudesta eri tilanteisiin ja muodostetaan yhteenveto.

4.2.1 Coynen menetelmä

Coynen vuonna 1995 julkaisema menetelmä roolien määrittelyyn lähtee liikkeelle tavoitteista [Coy95]. Onnistuneen roolien määrittelyn tavoitteena on määritellä roolit, käyttöoikeudet, rajoitteet ja hierarkiat. Coyne esittää tavoitteiden saavuttamiseksi 9 vaiheen menetelmää. Vaiheet ovat [Coy95]:

1. Käyttäjien suorittamien eri työtehtävien luettelointi. Luettelo muodostetaan verbi/objekti pareista kuten esimerkiksi ”hyväksy matkalasku”.
2. Edellä saadun luettelon ryhmittely yrityksessä olevien työntekijöiden toimenkuvan mukaan.
3. Ryhmittelyn tuloksena saatujen joukkojen nimeäminen substantiivilla, joista tulee alustavia rooleja.
4. Lyhyen kuvauksen kirjoittaminen alustaville rooleille.
5. Alustavien roolien vertailu ja päällekkäisyyksien poistaminen.
6. Minimikäyttöoikeuksien määritteleminen alustaville rooleille.
7. Käyttäjien toimien simulointi käyttämällä alustavia rooleja ja niihin liitettyjä käyttöoikeuksia.
8. Rajoitteiden määrittely rooleille. Rajoitteet tulisi löytyä yrityksen tietoturvapolitiikasta.
9. Roolien järjestely hierarkioihin. Hierarkioita voidaan muodostaa yritykselle relevanteista ominaisuuksista kuten työntekijöiden positioista ja organisaatiorakenteista.

Coynen esittämä menetelmä on ensimmäinen roolien määrittelyn problematiikkaan paneutuva artikkeli ja se on luonteeltaan abstrakti. Se ei ota kantaa esimerkiksi siihen, miten työtehtävät luetellaan ja miten löydetään minimijoukko käyttöoikeuksia kullekin roolille. Näistä puutteista johtuen Coynen menetelmä ei mielestäni sovellu suurille organisaatioille, joissa on lukuisia työtehtäviä, useita käyttäjiä ja suuri määrä erilaisia käyttöoikeuksia. Menetelmä sopinee paremmin pienemmille organisaatioille, joissa roolien määrittely pystytään suorittamaan kerralla läpi eikä esimerkiksi vaiheittaista siirtymistä tarvita.

4.2.2 Käyttötapaukset roolien määrittelyssä

Fernandez ja Hawkins lähestyvät roolien määrittelyn ongelmaa käyttötapausten (Use Cases) avulla. Vuonna 1997 julkaistussa artikkelissa he ehdottavat yksinkertaista metodia tarvittavien käyttöoikeuksien löytämiseksi eri rooleille. Menetelmä perustuu käyttötapauksiin, joita käytetään varsin yleisesti varsinkin oliopohjaisessa ohjelmointityössä ja vaatimusmäärittelyssä. [FeH97]

Käyttötapaukset kuvaavat käyttäjän ja järjestelmän vuorovaikutustilannetta. Ne ovat luonteeltaan semiformaaleja kuvauksia ja niitä voidaan esittää myös graafisessa muodossa kuten skenaariodiagrammeina. Perinteiset käyttötapaukset kuvaavat pelkästään toiminnallisia vaatimuksia. Tästä johtuen Fernandez ja Hawkins esittävät artikkelissaan laajennetun käyttötapausten määritelmän, joka mahdollistaa myös ei-toiminnallisten vaatimusten esittämisen. Ei-toiminnallisia vaatimuksia käytetään käyttöoikeuksien esittämiseen. [Feh97]

Kuvassa 8 on esimerkki tällaisesta laajennetusta käyttötapauksesta. Kuvaan on merkitty alleviivauksilla käyttötapausten laajennos, joka mahdollistaa tarvittavien käyttöoikeuksien esittämisen.

Otsikko: Materiaalin leikkaustilaus

Toimijat: Materiaaliosaston työntekijä

Esiehdot: Tilaus on hyväksytty. {tietoturva: materiaaliosaston työntekijä saa suorittaa materiaalin leikkauksen}

Kuvaus: Työntekijä suorittaa materiaalin leikkauksen haluttuihin osiin. [poikkeus: materiaalipula] {tieturvapoikkeus: Työntekijällä ei pääsyoikeutta}

Poikkeukset: Materiaalin puute – tilaus viivästyy {tietoturva: Työntekijällä ei pääsyoikeutta}

Jälkiehdot: Materiaali on leikattu {tietoturva: interaktio on kirjattu ylös}

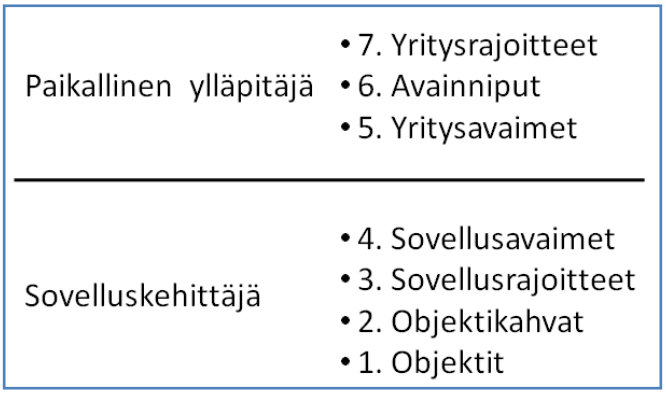
Kuva 8: Laajennettu käyttötapausesimerkki materiaalin tilauksesta [mukaillen FeH97].

Laajennetun käyttötapausten mallin ideana on tehdä jokaiselle työtehtävälle oma käyttötapaus. Näin saadusta käyttötapauslistasta voidaan määritellä tarvittavat oikeudet jokaiselle työtehtävälle erikseen. Menetelmä muistuttaa osittain Coynen esittämää menetelmää. Coynen menetelmän luetteloa vastaa käyttötapauslista. Käyttötapausmenetelmän eduksi voidaan lukea käyttöoikeuksien parempi hallinta. Menetelmä tukee vähäisem-

pien käyttöoikeuksien periaatetta eli jokaiseen työtehtävään tulee vain ja ainoastaan tarvittavat käyttöoikeudet. Laajennettu käyttötapausmalli ei ota kantaa roolien nimeämiseen eikä roolihierarkioiden luomiseen. Kirjoittajat toteavat itse, että käyttötapausten hyödyntäminen vaatii tarkennuksia ja lisätutkimusta. Menetelmän käyttöarvoa on vaikea arvioida. Käyttötapaukset sisältävät paljon hyödyllistä informaatiota helposti omaksuttavassa muodossa, joten sen käyttö voisi olla perusteltua eri työtehtävien dokumentoinnissa. Näin saatuja tietoja voitaisiin hyödyntää muuallakin organisaatiossa kuten esimerkiksi liiketoimintaprosessien kehityksessä.

4.2.3 Roolien määrittely hajautetuilla komponenteilla

Thomsen ja kumpp. [TOB98] lähestyvät roolien määrittelyn ongelmaa mielenkiintoisella teknisellä tavalla. He kuvaavat artikkelissaan hajautettuihin komponentteihin perustuvan RBAC-kehysrakenteen (framework), joka koostuu seitsemästä abstraktista kerroksesta (ks. kuva 9). Pääsynhallinnan toteutus on jaettu kahteen osaan. Sovelluskehittäjät toteuttavat sovelluskohtaisia pääsynhallintasääntöjä, jotka ovat usein monimutkaisia ja vaativat sovelluksen toiminnan syvempää tuntemusta. Paikalliset ylläpitäjät puolestaan tuntevat paikalliset tavat ja tietoturvakäytännöt, joiden perusteella he tekevät omia pääsynhallintaan vaikuttavia toimia.



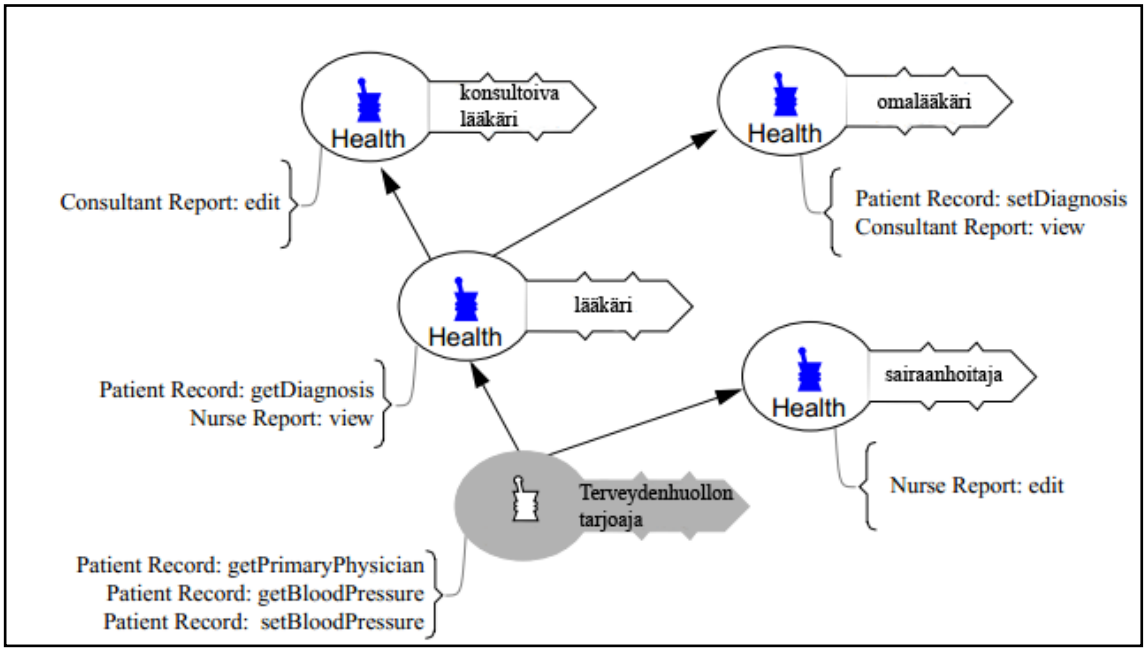
Kuva 9: Thomsenin ja kumpp. 7 abstraktia kerrosta [mukaillen TOB98].

Taso 1 sisältää objekteja, jotka vastaavat käytettävän hajautusteknologian komponentteja. Komponentteja käytetään kutsumalla niissä olevia julkisija metodeja. Pääsynhallinta toteutetaan kontrolloimalla sitä, kuka voi kutsua näitä metodeja.

Taso 2 koostuu kahvoista objekteihin. Usein objekteissa (eli komponenteissa) on useita metodeja, joita käytetään peräjälkeen jonkun tehtävän suorittamiseen. Objektikahvalla tarkoitetaan joukkoa, joka sisältää tehtäväkokonaisuuteen kuuluvat metodit. Näitä kahvoja voidaan sijoittaa eri rooleille tarvitsematta käydä metodeja läpi yksitellen.

Tasolla 3 ovat sovelluskohtaiset rajoitukset. Rajoitukset voivat liittyä muun muassa objektia käyttävän henkilöllisyyteen, objektiin itseensä tai aikaperusteisiin rajoituksiin. Rajoitukset liitetään tason 2 objektikahvoihin. Toisin sanoen ehtojen täytyy täyttyä ennen kuin pääsy objektikahvassa kuvattuihin metodeihin sallitaan.

Taso 4 kuvaa sovellusavaimet. Sovellusavain toimii kuten tavallinen avain. Tietyn avaimen antaminen käyttäjälle mahdollistaa avaimessa kuvatun resurssin käytön. RBAC-kehyksessä kuvatut avaimet ovat metodijoukkoja ja sovelluskohtaisia rajoitteita näihin metodeihin. Sovellusavaimen voidaan ajatella olevan sovelluskohtainen rooli. Sovellusavaimista voidaan muodostaa avainhierarkioita, joilla voidaan kuvata avainten keskinäisiä suhteita samalla tavalla kuin roolihierarkioissa (ks. kuva 10).

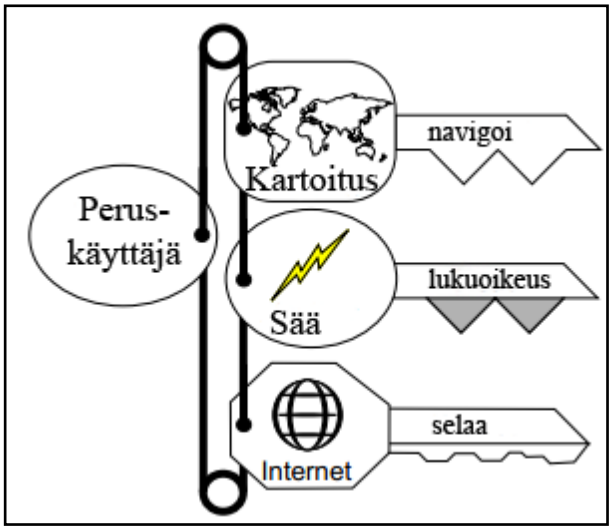


Kuva 10: Esimerkki avainhierarkiasta [mukaillen TOB98].

Taso 5 sisältää yritystason avaimet. Kun sovellus asennetaan verkkoon, kuvaus kaikista sen objekteista ja alustava sovellusavainlista tallennetaan hallinnointityökaluun. Paikalliset järjestelmänvalvojat antavat käyttöoikeuksia sovelluksiin sijoittamalla yritystason avaimia käyttäjille. Jokaista yritystason avainta kohden on vastaava sovellustason avain.

Käyttäjällä on pääsy yritysavaimessa kuvattuihin metodeihin edellyttäen, että avaimiin liitetyt sovelluskohtaiset rajoitteet tulevat täytetyiksi.

Taso 6 koostuu avainnipuista. Paikalliset ylläpitävät voivat koostaa yritystason avaimista koostuvia avainnippuja (ks. kuva 11). Avainnippujen tarkoituksena on toteuttaa paikallisia rajoitteita, jotka eivät ole selvillä sovelluksen tekohetkellä ja helpottaa hallinnointia.



Kuva 11: Esimerkki avainnipusta [mukaillen TOB98].

Taso 7 kuvaa yritystason rajoitteet. Yritystason rajoitteet liitetään avainnippuihin. Yritystason rajoitteilla on mahdollista vaikuttaa kaikkiin avainnipussa mainittuihin sovelluksiin yhdellä kertaa. Yritystason rajoitteet ovat samantyyppisiä kuin sovelluskohtaiset rajoitteet.

On huomattavaa, että Thomsenin ja kumpp. malli perustuu hajautettuihin komponentteihin. Tästä johtuen käytössä on oltava jokin hajautusta tukeva komponenttiteknologia kuten esimerkiksi CORBA tai Microsoftin COM/DCOM. Yhteistä edelle mainituille on rajapinnan kuvauskieli (Interface Definition Language), joka kuvaa kuinka komponenttia käytetään. Kirjoittajat esittävät artikkelien lopuksi prototyypin hallinnointisovelluksesta, joka lukee komponenttien IDL-kuvaukset ja mahdollistaa mm. avaimien, rajoitteiden ja objektikahvojen luomisen.

Thomsenin ja kumpp. esittämä menetelmä erosi paljon muista analysoimistani menetelmistä. Pääsynhallinnan jakaminen sovelluskehittäjien ja paikallisten ylläpitäjien kes-

ken on mielestäni hyvä idea. Yksinään kumpikaan ei ole riittävä, mutta yhdessä ne muodostavat varsin toimivan kokonaisuuden. Kerrosrakenne ja niiden sisällä oleva mahdollisuus hierarkioiden käyttöön luo joustavuutta ja yksinkertaistaa hallittavuutta. Komponenttiteknologiariippuvaisuus tuo mukanaan myös ongelmia. Olemassa olevien järjestelmien muuttaminen malliin sopivaksi on erittäin työlästä kuten kirjoittajatkin toteavat. Malli ei ota myöskään kantaa kuinka roolit tulisi määritellä, kuinka roolihierarkiat muodostetaan ja miten käyttöoikeudet pitäisi sijoittaa rooleihin. Malli on mielestäni enemmänkin tieteellinen konsepti kuin varteenotettava menetelmä kattavan pääsynhallinnan toteuttamiseksi.

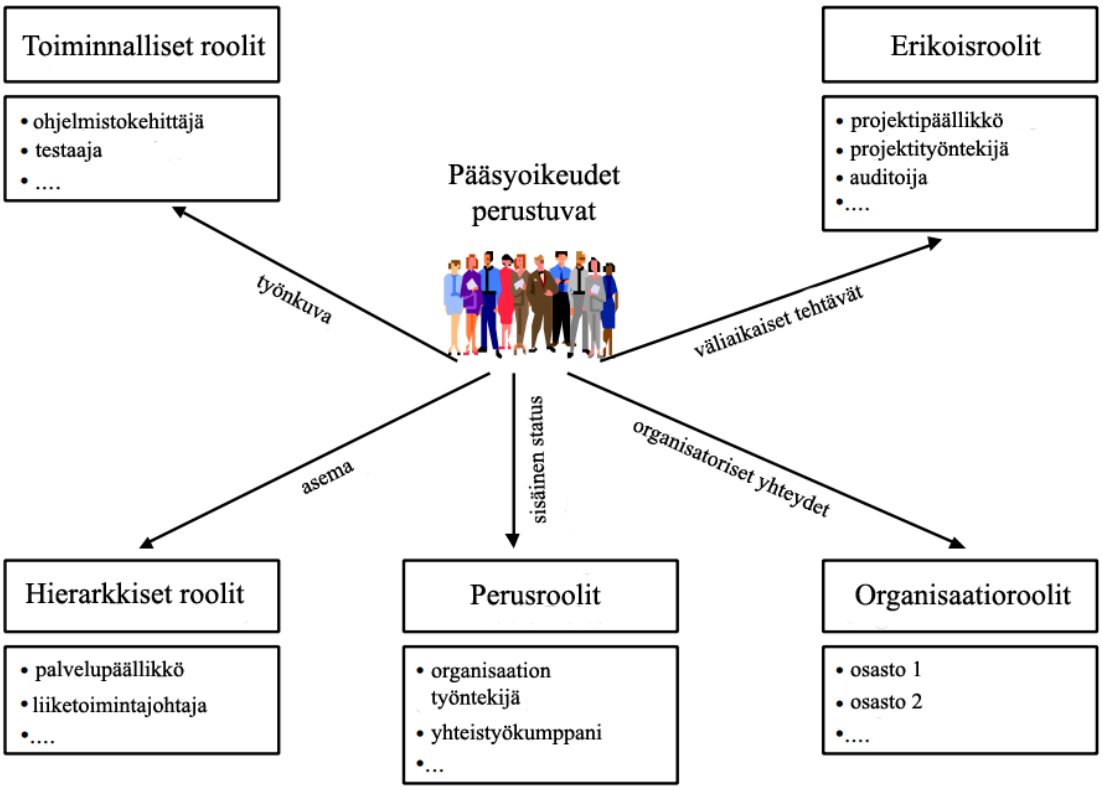
4.2.4 Prosessikeskeinen lähestymistapa

Roeckle ja kumppanit lähestyvät roolien määrittelyä prosessikeskeisesti vuonna 2000 julkaistussa artikkelissaan. Tapaustutkimuksen kohteena on yli 60000 loppukäyttäjän yritys, jonka tavoitteena on siirtyä roolipohjaiseen pääsynhallintaan, ja mahdollistaa kaikkien käyttäjien keskitetty hallinta järjestelmästä riippumatta. Aikaisemmin käyttäjien oikeuksia on hallinnoitu järjestelmäkohtaisesti, mikä on ollut kallista, aikavievää ja virhealtista. [RSW00]

Prosessikeskeisessä mallissa roolit jaotellaan viiteen eri luokkaan: toiminnalliset roolit, erikoisroolit, organisaatioroolit, perusroolit ja hierarkkiset roolit (ks. kuva 12). Rooliluokilla pyritään vähentämään roolienhallinnan tuomaa monimutkaisuutta, koska roolien suunnittelu ja hallinnointi vaikuttaa vain luokkaan johon rooli kuuluu. Näin ollen esimerkiksi liiketoimintaympäristön muutos vaikuttaa yleensä vain yhteen rooliluokkaan. Samoin toiminnallisten roolien etsimisen suuri työmäärä ei vaikuta muiden teknisimpien roolien suunnitteluun ja hallinnointiin. Eri rooliluokat kootaan lopuksi roolikatalogiin, joka sisältää kaikki yrityksessä käytössä olevat roolit. [RSW00]

Perusroolit, hierarkkiset roolit ja organisaatioroolit ovat määriteltävissä suhteellisen helposti käyttäen hyväksi esimerkiksi organisaatiokaavioita ja käytössä olevia titteleitä. Erikoisroolit ovat luonteeltaan väliaikaisia kuten projektiryhmän jäsen. Toiminnallisten roolien määrittely vaatii tuekseen roolienmäärittelyprosessin. Prosessin tarkoituksena on mm. vähentää tarvittavien roolien määrää sekä tehdä rooleista vankkoja, organisaatiora-

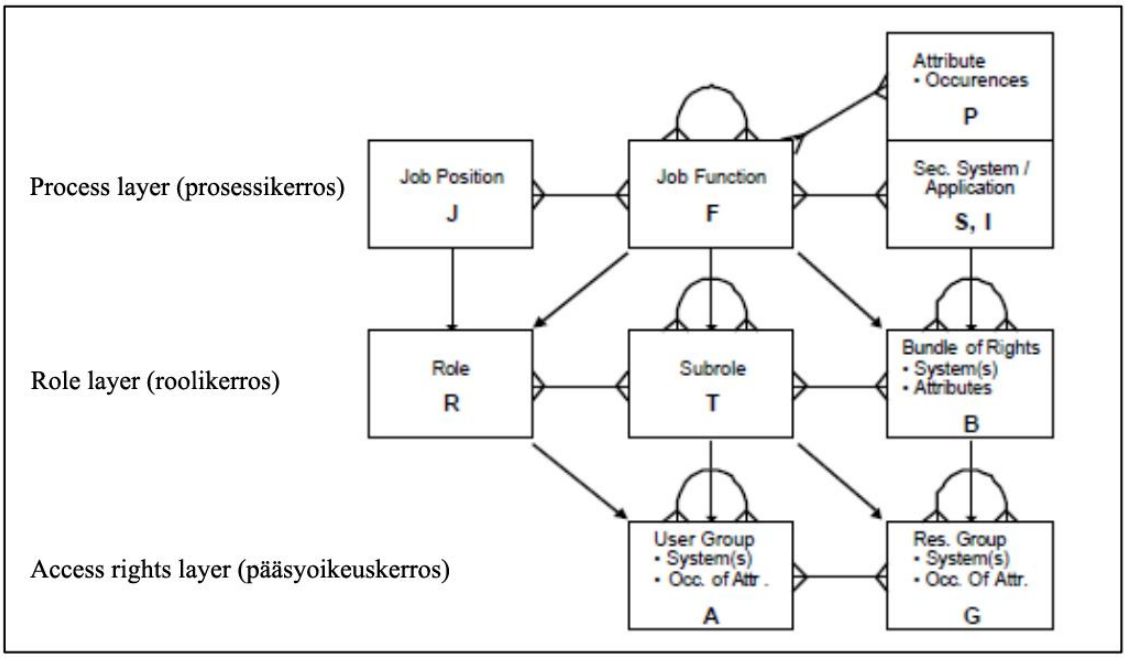
kenteen muutoksia kestäviä. Tätä tarkoitusta varten artikkelin kirjoittajat esittävät prosessikeskeistä roolien etsimistä. [RSW00]



Kuva 12: Roolien luokittelu Roecklen ja kumpp. mukaan [mukaillen RSW00].

Prosessikeskeisen roolien etsimisen ja määrittelyn menetelmä perustuu metamallin käsitteeseen. Metamalli koostuu kolmesta eri kerroksesta: prosessit, roolit ja pääsyoikeus (ks. kuva 13). Prosessikerros toimii rajapintana liiketoimintaprosessimalleihin. Se sisältää kuvaukset työtehtävistä, organisaatiorakenteesta, käytetyistä informaatiojärjestelmistä ja tietoturvakäytännöistä. Prosessikerros on muodostettavissa viidessä vaiheessa [RSW00]:

1. Etsi sopivat organisaatioyksiköt ja henkilöt, ja delegoi roolien etsimisvastuu heille.
2. Järjestä koulutus edellä mainituille henkilöille.
3. Etsi ja nimeä kaikki käytössä olevat työtehtävät, joissa käytetään apuna tietokonejärjestelmiä.
4. Etsi eri työpositioihin sopivat käyttöoikeudet, jotka vastaavat yksittäisen työntekijän tekemää työtä.
5. Lisää käytössä oleviin työtehtäviin niissä käytetyt tietojärjestelmät, tietoturvapalvelut ja muut työtehtävää kuvaavat attribuutit.



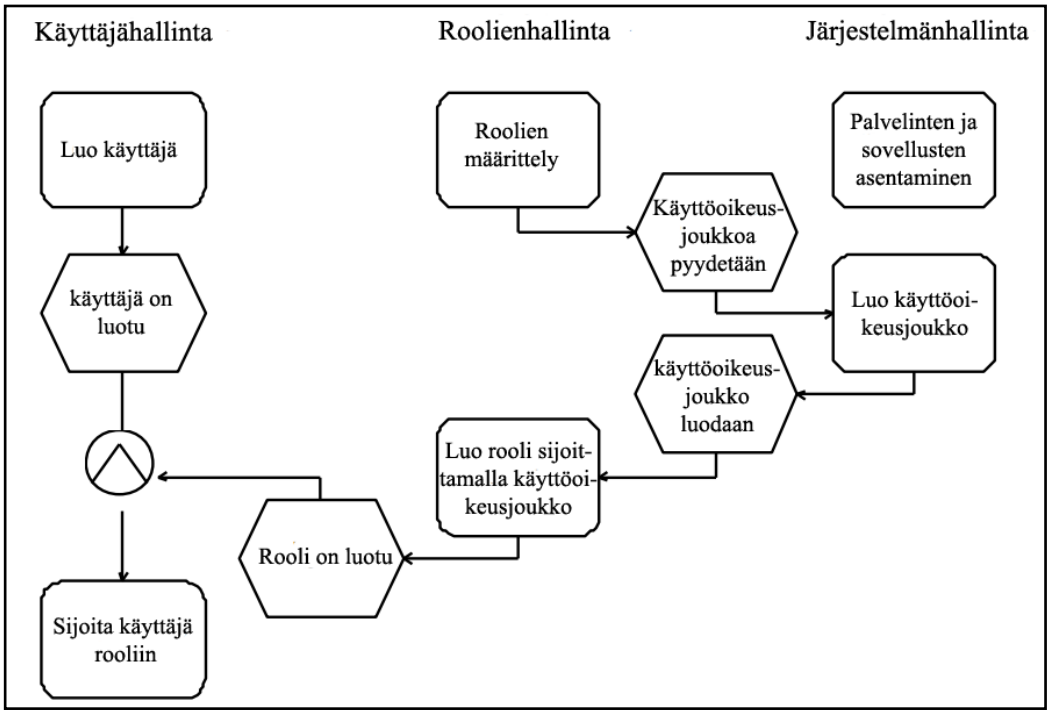
Kuva 13: Prosessikeskeisen roolien etsimisen metamalli [mukaillen RSW00].

Parhaassa tapauksessa edellä mainitut tiedot ovat jo valmiina yrityksen liiketoiminta-prosessimalleja kuvaavassa materiaalissa. Mikäli näin ei ole, joudutaan ne määrittelemään.

Kun prosessikerros on saatu valmiiksi, johdetaan näistä tiedoista rooli- ja pääsyoikeuskerrokset automaattisesti käyttäen tähän tarkoitukseen luotua algoritmia. Algoritmi päättelee mitkä eri prosessit voisivat toimia rooleina ottaen huomioon käytetyn tietoturvapolitiikan ja eri järjestelmien vaihtelevat pääsynhallintamekanismit. Valitettavasti Roeckle ja kumppanit eivät kerro käytetystä algoritmista artikkelissaan tämän tarkemmin, vaan toteavat julkaisevansa sen myöhemmässä vaiheessa. Kyseistä artikkelia ei löytynyt käyttämistäni tietokannoista.

Prosessikeskeinen malli on sidoksissa yrityksen yleiseen tietoturvan hallintaan (ks. kuva 14). Käyttäjähallinnassa tapahtuu useita toimenpiteitä viikoittain ja näiden toimenpiteiden yksinkertaistaminen ja nopeuttaminen roolipohjaisella pääsynhallinnalla on prosessikeskeisen mallin päätavoite. Paikallisten ylläpitäjien ei tarvitse tietää roolien teknisen toteutuksen yksityiskohtia, vaan he sijoittavat käyttäjät valmiiksi määriteltyihin ja kuvattuihin rooleihin. Tämä delegointi jakaa käyttäjähallinnan kuormaa järjestelmien yllä-

pitäjiltä paikallisille ylläpitäjille. Roolienhallinta pitää sisällään roolien määrityksen ja ylläpidon. Vaikka itse määrittely on tekninen, vaatii roolien määrittely vahvaa liiketoimintaosaamista ja järjestelmien teknisten yksityiskohtien ymmärtämistä. Järjestelmänhallinta koostuu palvelinten ja sovelluksien asentamisesta ja ylläpidosta. Prosessikeskeinen malli pyrkii yksinkertaistamaan järjestelmänhallinnan työnkuvia ja vähentämään työn monimutkaisuutta. [RSW00]



Kuva 14: Tietoturvan hallinnan prosessi [mukaillen RSW00].

Roeckle ja kumppanit toteavat, että roolien tulisi perustua yritysten toiminnallisiin rakenteisiin. Organisaatiorakenteet eivät yleensä ole stabiileja vaan ne muuttuvat ajan kuluessa. Liiketoimintaprosessit muuttuvat harvoin oleellisesti, josta johtuen niitä käyttämällä voidaan välttää roolien uudelleenmäärittelytyöltä. Roolien määrittely pelkkien työnkuvausten perusteella ei toiminut sekään hyvin. Suurimpana ongelmana oli tarvittavien konkreettisten käyttöoikeuksien puuttuminen työnkuvauksista. Tietoturvapolitiikka täytyy sitoa liiketoimintaprosesseihin eikä se voi perustua pelkästään tekniseen näkökulmaan. Liiketoimintaprosessien käyttö luo tarvittavaa tietämystä pääsynhallinnan tekniseen toteutukseen. Isoissa yrityksissä on myös tarpeellista käyttää roolien määritte-

lyyn ja etsimiseen tarkoitettuja ohjelmistoja, jotta työmäärät saadaan pidettyä kurissa ja dokumentaatio ja muutostenhallinta ajan tasalla. [RSW00]

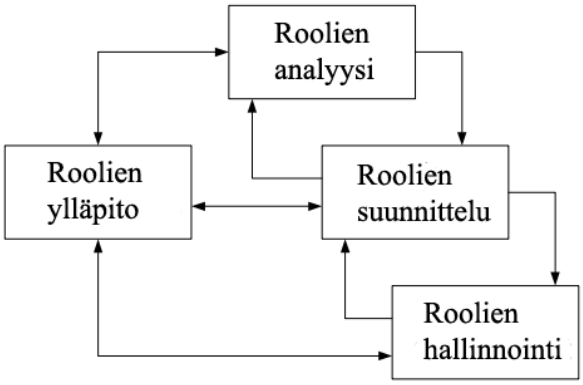
Prosessikeskeinen malli nojaa vahvasti liiketoimintaprosesseihin, joiden tarkalla kuvauksella voidaan johtaa tarvittavat roolit ja niihin liittyvät käyttöoikeudet. Menetelmän vahvuudeksi voidaan lukea roolien ja käyttöoikeuksien automaattinen generointi prosessikerroksessa olevista tiedoista. Luonnollisesti onnistuneeseen lopputulokseen vaikuttaa prosessikerroksessa olevan tiedon laatu. Roeckle ja kumppanit toteavat, että prosessikeskeinen lähestymistapa vaatii vielä lisätutkimusta ja tarkennuksia, jotta se olisi määritelty ja kuvattu tieteellisen tarkasti ja pätevästi. Joka tapauksessa esimerkkiyrityksen tapauksessa tulokset olivat varsin rohkaisevia.

4.2.5 Roolien määrittely elinkaarimallilla

Kern ja kumppanit lähestyvät roolien määrittelyä elinkaariajattelun näkökulmasta peilaamalla ohjelmistotuotannossa käytössä olevia menetelmiä [KKS02]. Perinteisesti ohjelmistokehitysprosessien malleina on käytetty vesiputous-, V-, ja spiraalimalleja. Mallit koostuvat eri vaiheista, jotka suoritetaan järjestyksessä. Esimerkiksi vesiputousmalli koostuu vaatimusmäärittelystä, suunnittelusta, toteutuksesta, testauksesta ja ylläpitovaiheista. Yhteistä perinteisille malleille on se, että ne olettavat ohjelmistolle asetettavien vaatimusten säilyvän suhteellisen samoina koko tuotantoprosessin ajan. Moderni ohjelmistokehitys lähtee liikkeelle puolestaan vaiheittaisen kehityksen pohjalta. Alustavan analyysin jälkeen luodaan malli, joka on niin täydellinen kuin mahdollista sen hetkisten tietojen perusteella. Tämä perusmalli jaetaan useaan osaan ja kutakin osaa suunnitellaan, toteutetaan ja testataan erikseen. Mallin eduiksi nähdään mm. osittaisen toiminnan saavuttamiseen tarvittava lyhyt aika ja helpompi muutoksiin sopeutuminen. Eri inkrementtien tuloksia voidaan esitellä tarpeen mukaan loppukäyttäjille, ja käyttää näin saatua arvokasta palautetta seuraavan inkrementin suunnittelun apuvälineenä. [KKS02]

Kernin ja kumppanien elinkaarimalli perustuu inkrementaaliseen malliin, joka koostuu neljästä vaiheesta: analyysi, suunnittelu, hallinnointi ja ylläpito (ks. kuva 15). Kuvassa esiintyvät nuolet kuvaavat iteratiivista prosessimallia, jossa on mahdollista liikkua eri vaiheiden välillä molempiin suuntiin. Eri vaiheita ei suoriteta peräkkäin, vaan alustavan roolijoukon määrittelemisen jälkeen roolien analysointi ja suunnittelu tapahtuvat inkre-

mentaalisesti sykleissä eri osa-alueilla. Tällaisia osa-alueita voivat olla esimerkiksi organisaation eri yksiköt. [KKS02]



Kuva 15: Roolin elinkaari [mukaillen KKS02].

Normaalisti valitaan jokin pienehkö osa-alue prototyypiksi, jonka avulla voidaan varmistua suunnittelun oikeellisuudesta ennen toteutuksen täysimittaista käyttöönottoa. Näin voidaan rajata suunnittelussa tapahtuvien virheiden esiintyminen pieneen osaan organisaatiota, eikä koko organisaation kattavaa roolirakennetta tarvitse suunnitella uudelleen. Valmiiksi toteutettujen osa-alueiden roolit voidaan myös ottaa käyttöön aikaisessa vaiheessa ja niistä saatuja kokemuksia ja roolirakenteita hyödyntää toisilla alueilla. Tämä tasaa roolien hallinnoinnista aiheutuvia työmääriä ja kustannuksia. [KKS02]

Elinkaarimallin analyysivaihe pitää sisällään roolien etsimisen kohdealueelta. Ideaalitapauksessa roolien etsiminen tulisi toteuttaa ylhäältä alas (Top-Down) ja alhaalta ylös (Bottom-Up) menetelmien yhtäaikaisella käytöllä. Top-Down–menetelmä perustuu organisaatiota kuvaavaan tietoon, joka on saatavilla sähköisessä muodossa tai tuotettavissa haastattelujen avustuksella. Tarvittavia tietoja voidaan jalostaa esimerkiksi organisaatiokaavioista, työnkuvauksista, liiketoimintaprosessien kuvauksista tai käytetyistä tietoturvakäytännöistä.

Bottom-Up–menetelmä lähtee liikkeelle vastakkaisesta suunnasta. Olemassa olevien järjestelmien käyttöoikeuksia tutkimalla voidaan muodostaa käyttöoikeusjoukkoja, jotka esiintyvät ryppäissä ja ovat näin ollen mahdollisia tulevia rooleja. Menetelmää kutsutaan kirjallisuudessa myös roolien louhimiseksi (Role Mining).

Edellä esitellyillä menetelmillä on omat hyvät ja huonot puolensa. Top-Down –menetelmä ei huomioi olemassa olevia käyttöoikeuksia, eikä Bottom-Up–menetelmä huomioi organisaatiorakenteita. Edellä esitettyjä ongelmia pyritään minimoimaan menetelmien yhtäaikaisella käytöllä.

Roolimallin toteuttaminen vaatii käytettyjen rakenteiden, kuten organisaatiorakenne, syntaksista ja semanttista kuvaamista käytetyn roolinhallintasovelluksen ymmärtämään muotoon. Tätä vaihetta kutsutaan elinkaarimallissa roolien suunnitteluksi. Suunnitteluvaiheessa toteutetaan tarvittavat roolihierarkiat, käyttäjä-rooli relaatiot ja rooli-käyttöoikeus relaatiot. Roolimallin automaattinen ylläpito vaatii toimiakseen algoritmin, joka suunnitellaan tässä vaiheessa. Tavoitteena on, että organisaatiomuutokset pystyttäisiin viemään roolimalliin automaattisesti. Aina tämä ei ole mahdollista, jolloin muutokset roolimalliin täytyy tehdä käsin. [KKS02]

Roolinhallintavaihe koostuu rutiininomaisista toimenpiteistä kuten uusien käyttäjien sijoittamista sopiviin rooleihin ja tiettyyn roolin kuuluvien käyttöoikeuksien hienosäätämistä. Apuna ja ohjenuorana käytetään suunnitteluvaiheessa syntynyttä dokumentaatiota ja määrittelyjä. [KKS02]

Suuret muutokset organisaatiorakenteessa tai esimerkiksi yritysten yhdistyminen voivat aiheuttaa isoja muutoksia käytössä olevaan roolimalliin. Tämä edellyttää käytettyjen konseptien uudelleenmäärittelyä samaan tapaan kuin roolien suunnitteluvaiheessa. Näitä muutoksia tehdään tarvittaessa ylläpitovaiheessa. Ylläpitovaihe eroaa suunnitteluvaiheesta siltä osin, että hyvin harvoin roolit täytyy määritellä aivan alusta asti uudelleen, vaan aikaisempia roolirakenteita voidaan hyödyntää myös uudessa mallissa. Ylläpito- ja suunnitteluvaihe ovat luonteeltaan haasteellisia, ja tämän vuoksi niissä tarvitaan yhteistyötä kohdealueen tuntevien ammattilaisten, systeemisuunnittelijoiden ja ylläpitäjien kesken. [KKS02]

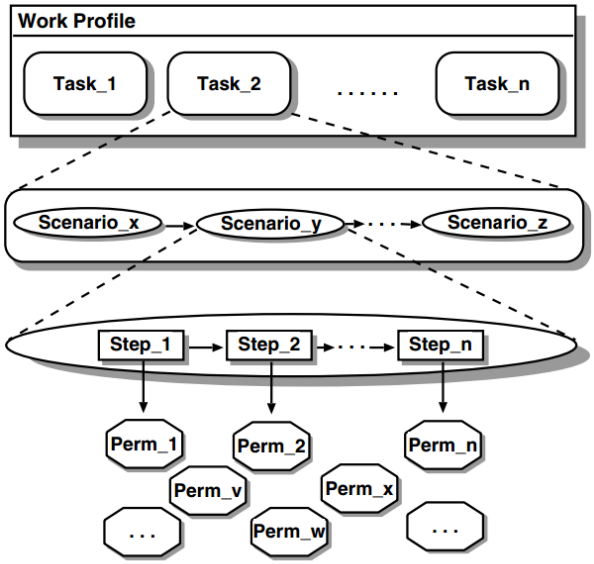
Kirjoittajat toteavat elinkaarimallin idean käytöstä roolien määrittelyssä olevan vielä kesken ja vaativan lisätutkimusta. Malli soveltunee paremmin suurille yrityksille, koska vaikka artikkelissa todetaan mallin olevan ohjelmistoriippumaton, toteavat kirjoittajat käyttäneensä mallin luomisessa suurille yrityksille tarkoitettua käyttöoikeuksienhallintaohjelmaa. Elinkaarimallin toiminta on esitetty varsin abstraktilla tasolla, eikä se ota

kantaa roolinmäärityksin vaikeisiin yksityiskohtiin kuten esimerkiksi miten käytännön tasolla muodostetaan toimiva RBAC-roolimalli. Tämä tietynlainen epätarkkuus johtunee varhaisesta kehitysasteesta, kuten kirjoittajatkin toteavat, mutta mielestäni myös mallista itsestään. Erilaiset mallit lähestyvät ongelmaa eri lähtökohdista, eikä niitä olekaan tarkoitettu kaikenkattaviksi kuvauksiksi. Roolien elinkaaren perustuva malli on piristävä ja peruslähtökohdiltaan hyvin erilainen verrattuna muihin tutkielmassani esitettyihin tapoihin. Samaan aikaan se kuitenkin hyödyntää hyväksi havaittuja ohjelmistotuotannon menetelmiä (inkrementaalien kehitys) ja roolien määrittelyn vakiintuneita käytäntöjä (Top-Down ja Bottom-Up -menetelmät).

4.2.6 Skenaarioperusteinen roolien määrittely

Neumann ja Strembeck esittävät vuonna 2002 julkaistussa artikkelissaan skenaarioperusteisen mallin, jonka avulla voidaan määritellä organisaation toiminnalliset roolit [NeS02]. Skenaariot ovat hyvin tunnettuja ja niitä käytetään paljon sovellustuotannossa. Skenaarioita käytetään kuvaamaan eri järjestelmien toimintaa ja ne helpottavat kommunikointia eri tahojen välillä. Tämä mahdollistaa myös ei-teknisten henkilöiden osallistumisen määrittelytyöhön. Neumannin ja Strembeckin mielestä roolien määrittelyyn liittyy vahva inhimillinen ja sosiaalinen tekijä, jonka johdosta skenaariot sopivat hyvin määrittelyprosessin suorittamiseen. Skenaarioperusteinen malli kattaa kaikki roolienmäärittelyn vaiheet ja se korostaa muutosten hallinnan tärkeyttä. Mallista on pyritty tekemään myös joustava, jotta sitä voisi käyttää erilaisissa organisaatioissa. [NeS02]

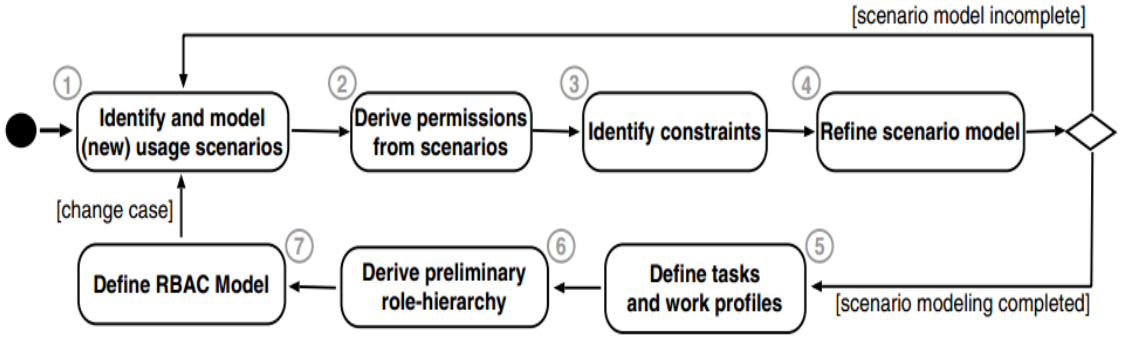
Skenaarioperusteinen malli koostuu työprofiileista (work profile), työtehtävistä (task), skenaarioista (scenario), askelista (step) ja käyttöoikeuksista (permission) (ks. kuva 16). Työprofiili pitää sisällään kaikki työtehtävät, joita yksittäinen työntekijä suorittaa työssään. Yksittäinen työtehtävä koostuu puolestaan työtehtävän suorittamisesta koostetuista skenaarioista. Jokainen skenaario on jaettavissa aliosiin, askeliin, jotka kuvaavat skenaarion osaa. Skenaarion osaan, eli askeleeseen, liitetään askeleen suorittamisessa tarvittavat käyttöoikeudet. [NeS02]



Kuva 16: Skenaariomallin rakenne [NeS02].

Esimerkiksi vakuutusvirkailijan eräs työtehtävä on vahinkoilmoitusten käsittely. Ilmoitusten käsittely on osa vakuutusvirkailijan työprofiilia. Työtehtävästä johdettu yksittäinen skenaario voisi olla autokolaritapauksen tietojen kirjaus vakuutusyhtiön tietojärjestelmiin. Skenaario kuvaa tässä tapauksessa sen, mitä eri järjestelmiä ja tietoja vakuutusvirkailija kirjaa tapauksen osalta ylös. Autokolaritietojen kirjausskenaario koostuu yksittäisistä askelista, jotka ovat esimerkiksi asiakkaan tietojen hakeminen asiakasrekisteristä ja autokolaritapauksen kirjaaminen vahinkotapaustietokantaan. Ensimmäinen askel edellyttää lukuoikeutta asiakasrekisteriin ja jälkimmäinen oikeutta lisätä uusi tietue vahinkotapaustietokantaan.

Skenaarioperusteinen roolien määrittelyprosessi on kuvattavissa seitsemän vaiheen avulla (ks. kuva 17).



Kuva 17: Skenaariomallin prosessikuvaus [NeS02].

Vaiheet ovat [NeS02]:

1. *Tunnista ja mallinna skenaariot*.
2. *Johda skenaarioissa tarvittavat käyttöoikeudet* työtehtävien ja askelten avulla. Käyttöoikeudet tallennetaan käyttöoikeuskatalogiin.
3. *Määritä rajoitteet*. Rajoitteita voivat olla esimerkiksi vastuiden eriyttämien ja aika- ja paikkasidonnaisuus. Rajoitteet tallennetaan rajoitekatalogiin.
4. *Jalosta skenaariomallia*. Samankaltaiset skenaariot pyritään yhdistämään ja skenaarioiden jakamista vielä pienempiin konkreettisiin aliskenaarioihin harkitaan.
5. *Määrittele työtehtävät ja -profiilit*. Sama skenaario voi kuulua useaan työtehtävään ja vastaavasti työtehtävä voi olla osana useampaa skenaariota.
6. *Määrittele alustava roolihierarkia*. Työprofiilien ja käyttöoikeuskatalogin avulla on mahdollista muodostaa alustava roolihierarkia puoliautomaattisesti.
7. *Määrittele RBAC-malli*. Mallin perustana on alustava roolihierarkia, käyttöoikeus- ja rajoitekatalogi. Päällekkäiset roolit poistetaan, uusia rooleja luodaan ja roolihierarkioita yhdistetään tai eriytetään tarvittaessa. Edellä kuvattuja toimenpiteitä toistetaan kunnes roolien määrittelijät toteavat konkreettisen mallin olevan riittävän hyvä.

Edellä esitetyn skenaariomallin prosessikuvan vaiheet 1-4 muodostavat syklin, jota suoritetaan kunnes skenaariomalli on valmis. Vasta tämän jälkeen siirrytään vaiheisiin 5-7. Tästä huolimatta vaiheita 1-7 on tarkoitus suorittaa iteratiivisesti ja inkrementaalisti, jossa jokainen iteraatiokierros johtaa uusiin päivitettyihin malleihin. Kun skenaariomalli on saatu valmiiksi (vaiheet 1-4), voidaan muutoksia, jotka parantavat mallin toimintaa lisätä vaiheessa 4, jossa jalostetaan skenaariomallia tai vaiheessa 7, jossa konkreettinen RBAC-malli on jo olemassa. Tämän jälkeen tarvittavat käyttöoikeudet johdetaan uudesta skenaariosta, skenaario liitetään yhteen tai useampaan työtehtävään ja RBAC-mallia päivitetään vastaamaan muutoksia.

Skenaarioperusteinen roolien määrittely mahdollistaa alustavan roolihierarkian luomisen puoliautomaattisesti. Jokaisesta työprofiilista luodaan ensin rooli, jolle annetaan kuvaava nimi. Koska työprofiileihin liitetyt työtehtävät koostuvat skenaarioista, jotka puolestaan rakentuvat askelista, joihin on liitetty tarvittavat käyttöoikeudet, voidaan roolin tarvitsemat käyttöoikeudet selvittää suoraan. Kun kaikista työprofiileista on luotu oma rooli, selvitetään roolien päällekkäisyydet. Tällä tarkoitetaan rooleja, joilla on täsmälleen samat käyttöoikeudet. Päällekkäisiä rooleja ei poisteta vaan ne merkitään myö-

hempää tarkastelua varten. Näin menetellään, koska roolien tarvitsemat käyttöoikeudet voivat muuttua myöhemmin ja niillä voi olla vain väliaikaisesti samat oikeudet. Toinen vaihtoehto on, että useammalle päällekkäiselle roolille muodostetaan yhteinen juniorirooli tai yhdestä päällekkäisestä roolista tulee toisen juniorirooli. [NeS02]

Roolihierarkian muodostamisen viimeisessä vaiheessa etsitään kaikki junioriroolit. Tämä suoritetaan etsimällä kaikki roolit (*r2*), joihin sijoitetut käyttöoikeudet ovat toisen roolin (*r1*) käyttöoikeuksien osajoukko. Toisin sanoen *r1* > *r2*. Tämän jälkeen muodostetaan perimisjärjestys roolien *r1* ja *r2* välille siten, että *r1* > *r2* on voimassa. Lopuksi päällekkäiset käyttöoikeudet poistetaan rooleista. Tällä tarkoitetaan käyttöoikeuksia, jotka ovat sijoitettuna rooliin suoraan sekä perittynä kyseisen roolin junioriroolilta. Kuvassa 18 esitetään edellä mainittu alustavan roolihierarkian muodostus pseudokoodilla. [NeS02]

```
for each work-profile {
  create role and assign permissions
  add role to allRoles
}
for each role1 in allRoles {
  for each role2 in allRoles {
    if {permissions of role1 = permissions of role2} {
      add role1 and role2 to potentiallyRedundantRoles
    }
    if {role1 > role2} {
      add role2 to juniorRoles(role1)
    }
  }
}
for each role in allRoles {
  if {juniorRoles(role) exists} {
    for each jrole1 in juniorRoles(role) {
      for each jrole2 in juniorRoles(role) {
        if {jrole1 > jrole2} {
         delete jrole2 from juniorRoles(role)
      }
    }
  }
}
for each role in allRoles {
  for each jrole in juniorRoles(role) {
    role addInheritanceRelationTo jrole
  }
  role remove redundant permissions
}
```

Kuva 18: Alustavan roolihierarkian muodostusalgoritmi pseudokoodina [NeS02].

Skenaarioperusteinen roolien määrittely on tutkituista menetelmistä kaikkein tunnetuin, koska se esitellään alan perusteoksissa ja siitä on johdettu eri variaatioita [CoD07; FKC07]. Kirjoittajat toteavat käyttäneensä mallia onnistuneesti kolmessa tapaustutkimuksessa ja neljäs on meneillään. Skenaariomalli on ollut perustana myös yhdysvaltalaiselle HL7-organisaatiolle (Health Level Seven), joka määrittelee terveydenhuollon standardeja ja edistää tietojärjestelmien kehittämistyötä. Eri HL7-määrityksiä on käytössä mm. Suomen terveydenhuollossa. Mielestäni mallin merkittävimmät oivallukset ovat skenaarioiden käytön mahdollistama kommunikaatio eri tahojen välillä, alustavan roolihierarkian puoliautomaattinen generointi sekä malliin sisäänrakennettu vastuiden eriyttäminen. Edellä mainituista varsinkin ensimmäinen on tärkeä, koska kuten kirjoittajat toteavat, roolien määrittely on vaatimusmäärittelyä ja siten hankalasti automatisoitavissa. Ihmisillä ja ihmisten välisellä kommunikaatiolla on näin ollen suuri merkitys onnistuneessa määrittelytyössä.

Suurimmaksi haasteeksi Neumann ja Strembeck toteavat skenaarioiden kattavuuden. On lähes mahdotonta mallintaa kaikki mahdolliset skenaariot. He toteavat ongelman olevan kuitenkin varsin tuttu ohjelmistotuotannossa. Esimerkiksi 100 prosentin testikattavuutta ohjelmistotestauksessa ei ole mahdollista saavuttaa, mikäli testattava ohjelma on ei-triviaali.

4.2.7 Tavoiteperusteinen roolien määrittely

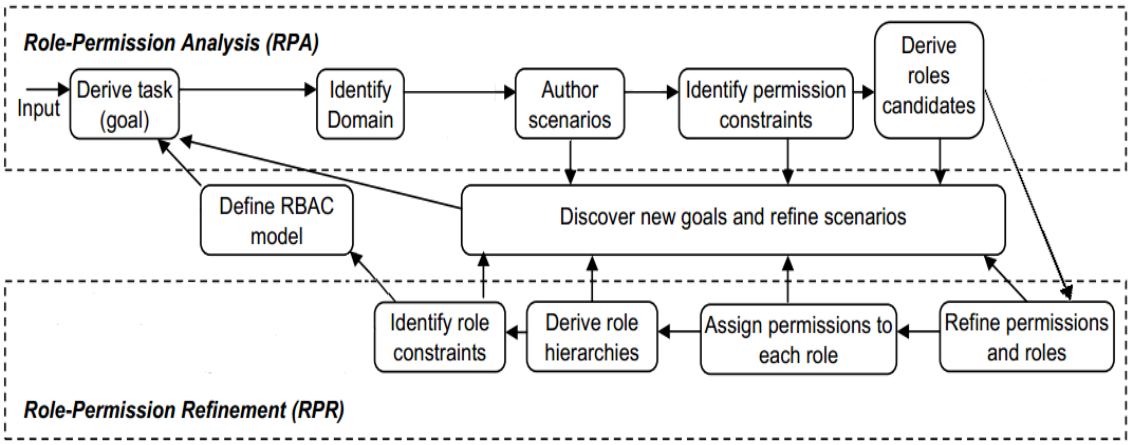
He ja Anton esittävät vuonna 2003 julkaistussa artikkelissaan, ”*A Framework for Modeling Privacy Requirements in Role Engineering*”, tavan ottaa huomion tietosuojatarpeet roolien määrittelyssä [HeA03]. Vaikka pääpaino artikkelissa on tietosuojan mallintamisessa, esitetään siinä myös tavoitelähtöinen (goal-oriented) roolien määrittelyprosessi, joka on riippumaton artikkelissa esitetystä tietosuojan mallintamisesta. Kirjoittajien esittämä tavoitelähtöinen roolien määrittelyprosessi hyödyntää aikaisemmin esitettyä skenaarioperusteista roolien määrittelyä. [HeA03]

Tavoitteet ovat yleensä abstrakteja ja deklaratiivisia. Niillä kuvataan esimerkiksi liiketoiminnan, organisaation tai järjestelmän tarkoitusta. Skenaariot ovat puolestaan konkreettisia, kerronnallisia ja proseduraalisia. Ne kuvaavat oikeita tilanteita käyttäen hyväksi esimerkkejä ja kuvainnollistamistapoja. Tavoitteita voidaan käyttää skenaarioiden

määrittelyssä ja skenaarioiden analysointi voi puolestaan helpottaa tavoitteiden määritystä. Tämän johdosta edellä mainittujen tapojen yhdistäminen on tehokas tapa saada esille ja johtaa vaatimuksia. Kuten kirjoittajat toteavat, roolien määrittely on vaatimusmäärittelyä. [HeA03]

He'n ja Antonin esittämä tavoitelähtöinen roolien määrittelyprosessi koostuu kahdesta vaiheesta: rooli-käyttöoikeus -analyysi (Role-Permission Analysis, RPA) ja roolikäyttöoikeus -täsmennys (Role-Permission Refinement, RPR). RPA-vaihe lähtee liikkeelle tavoite- ja skenaarioperusteisin tekniikoin, joilla analysoidaan liiketoimintaprosesseja ja -tehtäviä. Informaation lähteinä toimivat esimerkiksi liiketoimintaprosessikuvaukset, tietoturvakäytännöt ja vaatimusmäärittelydokumentti. [HeA03]

Kuvassa 19 havainnollistetaan RPA- ja RPR-vaiheiden toimintaa. Alkuperäisestä kuvasta poiketen siitä on poistettu yksityisyydensuojaan liittyvät vaiheet.



Kuva 19: Tavoitelähtöinen roolienmäärittelyprosessi [mukaillen HeA03].

Ensimmäisessä RPA-vaiheessa identifioidaan tehtävät. Tehtävällä tarkoitetaan toimia, joilla pyritään johonkin tavoitteeseen. Esimerkiksi "järjestä tapaaminen" tehtävän voidaan ajatella olevan kalenterisovelluksen tehtävä ja tehtävän tarkoituksena on järjestää tapaaminen. Kohdealueella (domain) tarkoitetaan tässä yhteydessä kalenterisovellusta. Tämän jälkeen työn alla olevasta tehtävästä koostetaan yksi tai useampi skenaario. Skenaario koostuu yhdestä tai useammasta tapahtumasta (event), joiden avulla johdetaan RBAC-mallin mukaiset alustavat käyttöoikeudet kyseiselle tehtävälle. On huomattavaa, että tavoitelähtöisessä mallissa skenaariot kuvataan hieman eri tavalla kuin skenaarioperusteisessa mallissa. Tästä huolimatta molempia tapoja käyttämällä on mahdollista joh-

taa tehtävissä tarvittavat käyttöoikeudet. Alustavien käyttöoikeuksien jälkeen muodostetaan kandidaattiroolit. Roolit saadaan selville listaamalla eri tehtävien toimijat (actors). RPA-vaihetta toistetaan kunnes kaikki tehtävät ovat läpikäyty. Tässä vaiheessa koottuna ovat kandidaattiroolit ja niihin sijoitetut alustavat käyttöoikeudet, joita tarvitaan RPR-vaiheessa. [HeA03]

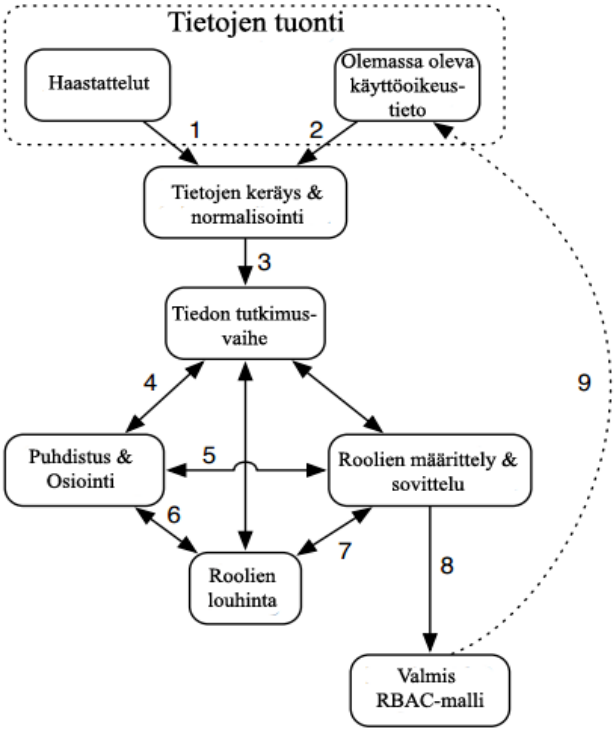
RPR-vaiheessa (ks. kuva 19) edellä määriteltyjä rooleja ja käyttöoikeuksia muokataan poistaen päällekkäisiä rooleja ja käyttöoikeuksia. Tämän jälkeen jäljelle jääneistä rooleista muodostetaan roolihierarkiat ja määritellään tarvittavat rajoitteet rooleille. Vaiheen lopputuloksena syntyy RBAC-malli. Mikäli uusia tavoitteita kohdataan, aloitetaan RPA-vaihe alusta, ja muuttuneet tai lisätyt kohdat päivittyvät RPA-vaihetta seuraavan RPR-vaiheen toimesta uuteen RBAC-malliin. [HeA03]

Tavoitelähtöinen roolien määrittely muistuttaa paljon skenaarioperusteista roolien määrittelyä. Molemmilla pystytään mallintamaan vain toiminnallisia rooleja ja ne käyttävät Top-Down–menetelmää roolien etsimiseen. Suurimpana erona edellä mainittujen menetelmien välillä pidän prosessikuvauksien yksityiskohtaisuutta. He'n ja Antonin menetelmä on yleisluontoisempi. Prosessi ei oteta kantaa esimerkiksi siihen miten roolihierarkiat muodostetaan ja yleensäkin eri vaiheisiin kuuluvia toimenpiteitä ei selitetä kovin tarkasti. Tämä johtunee osittain siitä, että tavoitelähtöinen roolien määrittelyprosessi on vielä alustava ja siihen on luvassa tarkennuksia. Toinen merkittävämpi syy on artikkelin tarkoitus. Artikkelin alkuperäisenä ideana on mallintaa eri tietosuojakäytäntöjä mukaan roolipohjaiseen pääsynhallintaan. Aiheen laajuudesta ja tutkielmani tarkoituksen johdosta rajasin nämä kuitenkin tutkielman ulkopuolelle.

4.2.8 Integroitu roolien määrittelyprosessi

Giblin ja kumppanit lähestyvät roolien määrittelyä teknisestä näkökulmasta vuonna 2010 julkaistussa artikkelissaan ”*Towards an Integrated Approach to Role Engineering*” [GGM10]. He esittävät artikkelissaan roolien määrittelyprosessin, joka hyödyntää useita eri tapoja toimivan RBAC-mallin muodostamiseksi. Artikkelissa kuvataan myös kirjoittajien luomaa kaupallista sovellusta, joka demonstroi integroidun menetelmän toimintaa. Integroidun menetelmän periaatteita voi kuitenkin käyttää ilman ko. sovellusta.

Giblin ja kumpp. esittävät RBAC-järjestelmän toteuttamisen koostuvan prosessista, jota iteroidaan useita kertoja (ks. kuva 20). Prosessin vaiheet ovat: tiedon keräys, tiedon tutkiminen, tiedon puhdistus ja osiointi, roolien louhinta, roolien määrittely ja sovittelu. Edellä esitetty järjestys on luonnollinen, vaikka vaiheet voidaan suorittaa myös toisessa järjestyksessä.



Kuva 20: Integroidun roolien määrittelyn työnkulku [mukaillen GGM10].

Tiedon keräysvaihe voidaan jakaa käyttöoikeustietojen ja organisaatiodatan keräämisen. Näitä tietoja saadaan niin ihmisiltä kuin tietojärjestelmistä suoraan. Yleensä käytössä olevat käyttöoikeustiedot eri järjestelmistä noudetaan tähän tehtävään tarkoitettua sovellusta hyödyntäen. Organisaatiodatan kerääminen on työläämpää kuin käyttöoikeustietojen. Usein kaikki roolien määrittelyn kannalta merkittävä organisaatiodata ei ole sähköisessä muodossa, vaan se tulee jalostaa esiin eri työntekijöitä haastattelemalla. Haastattelija kysyy työntekijöiltä tietoja käytetyistä tietojärjestelmistä, työnkuvista ja työtehtävistä. Haastatteluita käytetään luomaan liiketoimintakontekstia eri käyttäjä-käyttöoikeus –relaatioille. Toisin sanoen, niitä voidaan käyttää perustelemaan miksi jokin tietty käyttöoikeus täytyy olla tietyllä käyttäjällä liiketoiminnallisista lähtökohdista. [GGM10]

Tiedon tutkimusvaihe koostuu kerätyn aineiston analysoinnista. Käyttäjä-, käyttöoikeus- ja attribuuttiryppäitä on mahdollista etsiä hyödyntämällä tilastotieteen menetelmiä. Näitä ovat esimerkiksi matriisihajotelmat, klusterointi ja faktorianalyysi. Kerättyä aineistoa voidaan myös visualisoida, joka voi tuoda esiin roolien määrittelyn kannalta kiinnostavia malleja. Luokitteluasteikollisen tai binääristen tietojen, kuten esimerkiksi käyttäjien, käyttöoikeuksien ja roolien jakautumista, voidaan kuvata frekvensseillä ja histogrammeilla. Edellä mainittujen matemaattisten menetelmien ja organisaatiodatan avulla voidaan perustella mistä käyttäjä-käyttöoikeus–pareista kannattaisi luoda uusi rooli. [GGM10]

Tiedon puhdistus- ja osiointivaiheessa alustavia roolimäärityksiä ja niihin kuuluvia käyttöoikeuksia siistitään. Roolien määrittelijät pyrkivät yleensä 80/20-sääntöön, jolla tarkoitetaan sitä, että 80 prosenttia pääsynhallinnasta katetaan rooleilla ja loput 20 prosenttia ovat poikkeuksia, jotka käsitellään perinteisillä pääsynhallintamalleilla. Tätä tarkoitusta varten on kehitetty algoritmeja, jotka osaavat valita kaikkein sopivimman 80 prosentin osan datasta hoidettavaksi roolipohjaisella pääsynhallinnalla. Osioinnilla tarkoitetaan roolihierarkian jakamista pienempiin osiin organisatorista tai hallinnollisista syistä. Esimerkiksi saman roolin käyttämisen eri tulosyksiköissä tai maantieteellisissä sijainneissa voi olla kiellettyä. [GGM10]

Roolien louhintavaiheessa suoritetaan roolien ja roolihierarkioiden automaattinen etsintäprosessi, joka sopii parhaiten käyttäjä-käyttöoikeus–sijoituksille. Roolien louhinnalla tarkoitetaan erilaisia algoritmeja, jotka käyvät läpi olemassa olevia käyttöoikeustietoja ja esittävät niiden pohjalta roolijakoa, joka on mahdollisimman kattava. Paras sopivuus on subjektiivinen ominaisuus, joka riippuu ympäristön yksityiskohdista. Tiedonlouhintateknologiasta riippuen louhinnan tukena käytetään myös organisaatiodataa. Ideaalitapauksessa roolien määrittelijät voivat määritellä eri kriteereitä roolien louhintaan kuten sallittujen roolien määrän, roolihierarkian syvyyden ja esimääriteltyjä rooleja, jotka otetaan mukaan louhintaan. Louhinnan tulos voidaan esittää kahdessa eri muodossa. Ensimmäinen muoto kattaa täydellisen RBAC-mallin, johon kuuluvat roolit, roolihierarkiat ja mahdolliset käyttäjä-käyttöoikeus–parit, joille ei ole määriteltyä roolia. Jälkimmäinen muoto etsii malleja käyttäjä-käyttöoikeus–pareista ja palauttaa kandidaattirooleja

yksi kerrallaan kunnes jokin tietty osa käyttäjä-käyttöoikeus–pareista on läpikäyty. [GGM10]

Roolien määrittely- ja sovitteluvaiheessa roolit luodaan hyödyntämällä mm. organisaatiodataa. Joidenkin roolien oikeutus voi perustua lakiin, säädöksiin tai organisaation käytäntöihin. Tällaiset roolit ovat helppo esimääritellä. Roolien määrittelijän täytyy ottaa huomioon myös louhinnalla määritellyt roolit ja muodostaa niistä järjestelmärooleja mikäli ne soveltuvat kokonaisuuteen (ks. luku 2.4). Roolien määrittely ja sovittelu vaiheelle on tyypillistä, että rooleja hylätään, muokataan, yhdistellään ja jaetaan osiin tarpeen mukaan. Louhitulle rooleille annetaan tässä vaiheessa myös kuvaavat nimet, koska tällä hetkellä yksikään louhinta-algoritmi ei osaa nimetä louhimiansa rooleja merkityksellisesti. Vaiheen viimeisessä osassa tarkistetaan rajoitteiden oikeellisuus. Näitä ovat esimerkiksi vastuiden eriyttäminen ja muut operatiiviset rajoitteet. [GGM10]

Integroitu menetelmä on uusin tutkimistani roolien määrittelyprosesseista. Se yhdistelee alan uusimpia menetelmiä kuten roolien louhintaa, tiedon visualisointia ja eri tilastollisia menetelmiä. Rooleja etsitään sekä organisaatiodatasta (Top-Down) että louhimalla (Bottom-Up). Menetelmä nojaa vahvasti tietokoneiden laskentakapasiteetin valjastamiseen roolien määrittelijän avuksi.

Menetelmä on mielestäni varsin tehokas, mutta se myös edellyttää käyttäjältään paljon. Se soveltunee parhaiten roolin määrittelyn ammattilaisille, joilla on jo kokemusta ja näkemystä useista roolien määrittelyprojekteista. Integroidun mallin esittämiä tekniikoita voidaan tällöin käyttää päätöksenteon tukena, ja se mahdollistaa myös eri toteutusvaihtoehtojen simuloinnin. Kokemattomissa käsissä integroitu menetelmä voi olla arvaamaton, ja työkalu voi alkaa viedä määrittelijää, vaikka asian pitäisi olla toisin päin. Korostunut tekninen puoli jättää hieman taka-alalle roolien määrittelyn sosiaalisen puolen, joka on myös tärkeää määrittelytyössä. Malli soveltunee parhaiten suurille organisaatioille, joissa on tarpeeksi dataa eri analyysimenetelmien käyttöön. Pienemmissä organisaatioissa ei tällaista tarvetta ole, vaikka jotain yksittäisiä integroidun menetelmän tekniikoita voisi olla perusteltua käyttää. Näitä ovat esimerkiksi olemassa olevien käyttäjä-käyttöoikeus–parien louhiminen ja haastattelut. Menetelmä oli mielestäni tutkituista menetelmistä monimutkaisin ja vaativin edellyttäen mm. tilastotieteen perusteiden ja tiedon louhinnan periaatteiden tuntemista.

4.3 Vertailukriteerit ja menetelmien arviointi

Tutkitut roolien määrittelymenetelmät ja prosessit olivat hyvin erilaisia ja ne lähestyivät roolien määrittelyn kenttää hyvin erilaisista lähtökohdista ja painottivat mahdollisesti jotain tiettyä roolin määrittelyn aluetta. Tästä syystä menetelmien keskinäinen vertailu objektiivisesti oli varsin haastavaa.

Tutkimuksessa käytettiin vertailukriteerien pohjana Coynen esittämää määritystä roolien määrittelyprosessista, joka sisältää roolien, käyttöoikeuksien, rajoitteiden ja roolihierarkioiden määrityksen. Jotta vertailulla olisi enemmän käytännön hyötyä, käytin kriteereinä myös menetelmän soveltuvuutta erikokoisille yrityksille (P=pieni, K=keskisuuri, S=suuri) ja käytettyä roolien etsimissuuntaa (T=Top-Down, B=Bottom-Up, H=hybridi). Etsimissuunnalla on merkitystä, jos esimerkiksi organisaatiokuvauksia ja liiketoimintaprosesseja ei ole dokumentoitu kunnolla tai käyttöoikeuksia ei ole annettu tarkoituksenmukaisesti.

Taulukossa 4 vertaillaan eri menetelmiä edellä mainittujen kriteereiden perusteella. Jotta käytetty menetelmä tukisi jotain edellä mainituista kriteereistä, täytyy menetelmän sisältää ohjeita tai muita apukeinoja kyseisen kriteerin määrittelyyn. Esimerkiksi menetelmän ohje ”määrittele käyttöoikeudet” ei ole riittävä ohje, jotta kyseinen menetelmä tukisi kyseistä kriteeriä. ”Johda käyttöoikeudet määriteltyjen skenaarioiden tapahtumista” puolestaan on riittävä ohje, koska se sisältää konkreettisen apukeinon käyttöoikeuksien määrittelyyn. Näin toimimalla eri menetelmät saadaan paremmin vertailukelpoiseksi ja ylimalkaiset ohjeet saadaan karsittua sekoittamasta vertailun tuloksia. Eri kriteerien tarkempi laadullinen arviointi on hyvin subjektiivista ja vaikeasti mitattavissa, minkä johdosta olen jättänyt sen vertailun ulkopuolelle.

Taulukko 4: Roolien määrittelymenetelmien vertailu

	Coyne	Use Cases	Komponentti	Prosessi	Elinkaari	Skenaario	Tavoite	Integroitu
rooli	X	X		X		X	X	X
käyttöoikeus		X	X			X	X	X
roolihierarkia				X		X		X
rajoitteet						X	X	X
etsimissuunta	T	T		T	H	T	T	H
yrityskoko	P	P,K		S	K,S	P,K,S	P,K,S	K,S

etsimissuunta:**T**=Top-Down -menetelmä, **H**=hybridimenetelmä yrityskoko:**P**=pieni, **K**=keskisuuri, **S**=suuri

Coynen yhdeksän kohdan roolien määrittelyohje opastaa kuinka roolit ovat löydettävissä. Muiden kriteereiden osalta tällaisia ohjeita ei anneta. Menetelmän abstraktisuus ja eri kriteereiden täyttämättä jääminen herättää aiheellisen epäilyksen kannattaako sitä käyttää. Coynen menetelmän puolesta täytyy sanoa se, että siinä esitettyjä ideoita on nähtävissä myöhemmin julkaistuissa laajemmissa menetelmissä. Esimerkiksi skenaarioista on nähtävissä verbi-objekti–parien idean jatkojalostus.

Käyttötapaukset-menetelmä tukee roolien ja käyttöoikeuksien etsimisessä. Roolihierarkioiden ja rajoitteiden määrittämiseen se ei tarjoa keinoja. Käyttötapauksia käytetään varsin paljon eri alueilla, joten menetelmän käyttö voi olla perusteltua, mikäli ne ovat entuudestaan tuttuja. Koska menetelmä keskittyy lähinnä käyttöoikeuksiin eikä se kata kaikkia roolien määrittelyn vaiheita, on aiheellista olettaa, että roolien määrittelyprosessi tarvitsee tuekseen myös jonkun toisen menetelmän.

Hajautettuihin komponentteihin perustuva malli on omalaatuinen. Se auttaa kiistattomasti käyttöoikeuksien määrityksessä ja niiden sijoittamisessa eri rooleihin. Menetelmä on kuvattu kuitenkin teknisistä lähtökohdista, eikä sen kuvauksesta ilmene kuinka se auttaa valitsemieni muiden kriteereiden määrittelyssä. Tästä syystä en kyennyt päätte-

lemään menetelmän käyttämää roolien etsimissuuntaa enkä ole varma minkä kokoiset yritykset sen käytöstä hyötyisivät.

Prosessikeskeinen lähestymistapa tukee roolien ja roolihierarkioiden muodostamista. Tuki on kuitenkin hieman epäselvää, koska se nojaa algoritmiin, jota ei esitellä artikkelissa tarkemmin. Prosessikuvauksesta saa kuitenkin mielestäni tarpeeksi tietoja, jotta edellä mainitut kriteerit täyttyvät. Vaikka menetelmä kattaa myös käyttöoikeudet ja rajoitteet, niiden määrittäminen jäi jokseenkin epäselväksi, minkä johdosta kyseessä olevat kriteerit eivät täyttyneet. Rakenteensa johdosta ja tapaustutkimuksessa käytetyn yrityksen koosta voidaan päätellä menetelmän soveltuvan parhaiten suurille yrityksille.

Elinkaarimallin kuvauksesta voidaan lukea suoraan mallin käyttävän roolien etsimisessä sekä Top-Down että Bottom-Up–menetelmiä. Soveltuvuus keskisuurille ja suurille yrityksille käy mielestäni ilmi menetelmän kokeilussa käytetystä käyttäjienhallintasovelluksesta. Muut käyttämäni kriteerit jäivät avoimiksi mallin keskeneräisyydestä johtuen.

Skenaariomalli on tutkimistani menetelmistä kattavin. Se kuvaa varsin yksityiskohtaisella tasolla kunkin roolien määrittelyn vaiheen ja kattaa jokaisen valitsemani kriteerin. Tunnettavuuden ja prosessin tarkan kuvauksen ansiosta se soveltunee kaiken kokoisille yrityksille.

Tavoitelähtöinen roolien määrittelyprosessi muistuttaa paljon skenaarioperusteista roolien määrittelyprosessia. Tästä syystä eroja skenaarioperusteiseen roolien määrittelyyn ilmeni vain roolihierarkian muodostamisessa. Tätä vaihetta ei kuvattu riittävällä tarkkuudella, jotta ko. kriteeri olisi täyttynyt.

Integroitu menetelmä tukee kaikkia valitsemiani kriteereitä. Tilastollisten ja roolinlouhintatekniikoiden luonteesta johtuen sen voidaan sanoa toimivan sitä paremmin mitä enemmän käytettävää dataa on tarjolla. Tästä syystä soveltuvuus keskikokoisille ja suurille yrityksille on hyvä ja vastaavasti pienille heikko.

4.4 Yhteenveto tutkituista menetelmistä

Suorittamani vertailun perusteella näyttäisi vahvasti siltä, että erityisesti keskisuurissa ja suurissa yrityksissä kannattaisi käyttää skenaariopohjaista roolien määrittelyprosessia.

Sen toimivuutta on testattu ja koeteltu useammassa tapaustutkimuksessa, prosessi on kuvattu riittävällä tarkkuudella ja menetelmän toimivuus on tunnustettu alan kirjallisuudessa. Skenaarioperusteista mallia muistuttava tavoitelähtöinen lähestymistapa voi olla myös sopiva valinta varsinkin, jos organisaatiossa on käytetty tavoitelähtöistä vaatimusmäärittelyä. Suurille yrityksille tarkoitetun prosessikeskeisen mallin ongelmana pidän sen käyttämän algoritmin toimintaperiaatetta. Algoritmin toimintaa kuvattiin vain muutamalla sanalla, minkä johdosta menetelmän laatua oli vaikea arvioida.

Pienissä organisaatioissa menetelmän valinnalla ei ole niin suurta merkitystä. Käyttäjiä, rooleja ja käyttöoikeuksia niissä on yleensä varsin vähän, minkä johdosta roolien määrittelyprosessi on suoraviivaisempaa ja helpompaa. Tästä syystä menetelmän valinnan tulisi mielestäni perustua enemmän määrittelijöiden omin mieltymyksiin kuin pelkästään eri menetelmien tuomiin etuihin.

Integroitu menetelmä näyttää ensisilmäykseltä varsin hyvältä vaihtoehdolta. Menetelmän käyttäminen edellyttää kuitenkin varsin laajan ja erikoistuneen sovelluksen käyttämistä. Tällöin on vaarana, että määrittelytyö tulee liian riippuvaiseksi käytetystä sovelluksesta ja sen ominaisuuksista. Mikäli roolien määrittelijällä on tarpeeksi kompetenssia ja kokemusta, näkisin kuitenkin menetelmän käytön perusteltuna ainakin hyvin monimutkaisissa ympäristöissä. Mikään ei estä myöskään käyttämästä vain osaa menetelmässä kuvattuja tapoja. Tutkimustuloksia menetelmän toimivuudesta kuitenkin tarvitaan.

Coynen esittämää menetelmää en voi suositella, koska se kuvaa roolien määrittelyprosessin vain hyvin karkealla tasolla. Näin ollen sen käytännön hyöty roolien määrittelytyöhön jää alhaiseksi. Käyttötapauksien hyödyntäminen auttaa käyttöoikeuksien etsimisessä ja mallintamisessa. Niiden laaja levinneisyys ja tunnettavuus puoltavat menetelmän käyttöä osana määrittelytyötä. Yksinään menetelmä ei ole riittävä.

Komponentteihin perustuvaa mallia ja elinkaarimallia en käyttäisi käytännön roolien määrittelytyössä. Ensiksi mainittu kuvaa vain mallin teknisen rakenteen eikä näin ollen sovellu sellaisenaan roolien määrittelytyön apuvälineeksi. Mallin rajoittuneisuus komponenttiteknologian piiriin vähentää sen käyttömahdollisuuksia kattavana pääsynhallintamallina. Elinkaarimallin hyödyntämisen ongelmana näen mallin keskeneräisyyden.

Malli lähestyy kuitenkin roolien määrittelyä poikkeavasta näkökulmasta ja tuo mukanaan uusia ideoita. Tästä syystä mallin toiminnan tarkentuminen voi myöhemmässä vaiheessa muuttaa tilannetta.

5 POHDINTA

Luvussa kaksi tutkittiin pääsynhallintaa ja erityisesti roolipohjaista pääsynhallintaa. Pääsynhallinnan tarkoitus on rajoittaa mitä kaikkea käyttäjä pystyy tekemään tietojärjestelmässä. Sen toiminnan laadukkuutta voidaan arvioida tutkimalla siihen kohdistuvia tietoturvariskejä, jotka ovat jaettavissa kolmeen kategoriaan: luottamuksellisuus, eheys ja saatavuus. Roolipohjainen pääsynhallinta on yksi tapa toteuttaa yrityksen tai organisaation pääsynhallintatarpeet. Roolipohjaisen pääsynhallinnan perusideana on sijoittaa käyttäjien tarvitsemat käyttöoikeudet rooleihin, sen sijaan, että käyttöoikeudet annettaisiin suoraan käyttäjille. Käyttäjät saavat tarvitsemansa käyttöoikeudet rooliin sijoituksella. Roolipohjaisuuden on todettu mm. vähentävän hallinnointikustannuksia ja parantavan tietoturvaa. Tästä syystä entistä useampi yritys tai organisaatio on ottanut sen käyttöönsä.

Luvussa kolme kuvasin suorittamani systemaattisen kirjallisuuskatsauksen aineiston hankinta- ja analysointivaiheet. Katsauksen tarkoituksena oli tutkia ja vertailla eri roolin määrittelymenetelmiä ja tehdä vertailun pohjalta päätelmiä niiden sopivuudesta käytännön roolin määrittelytyöhön. Roolipohjaista pääsynhallintaa on tutkittu varsin paljon, mutta roolien määrittelyn osa-aluetta ei. Tästä syystä tutkimusongelmani kannalta mielenkiintoisia artikkeleita oli paikoin haastavaa seuloa esiin. Systemaattiseen kirjallisuuskatsaukseen valikoitui lopulta 8 tieteellistä artikkelia.

Luvussa neljä esitin suorittamani systemaattisen kirjallisuuskatsauksen tulokset. Vertailun lähtökohtana käytin kuutta eri kriteeriä. Tutkitut menetelmät erosivat toisistaan paljon niin ominaisuuksien kuin käyttökelpoisuudenkin osalta. Skenaarioperusteinen ja integroiva menetelmä tukivat jokaista arvioinnin kohteena olevaa ominaisuutta. Muissa menetelmissä oli havaittavissa selviä puutteita tältä osin.

Tutkimuksen tavoitteena oli etsiä tieteellisessä kirjallisuudessa esiintyvät menetelmät, jotka tukisivat roolien määrittelyprosessia. Aihe on merkittävä, koska roolien määrittely on yksi tärkeimmistä vaiheista roolipohjaiseen pääsynhallintaan siirtymisessä. Samalla se on myös työläin ja kallein. Tämän johdosta oikean menetelmän valinta on ensiarvoisen tärkeää.

Tieteellisestä kirjallisuudesta löytyi kahdeksan eri menetelmää, jotka tukevat roolien määrittelytyötä. Nämä olivat Coynen perusmenetelmä, käyttötapauksiin perustuva menetelmä, komponenttiteknologiaa hyödyntävä menetelmä, prosessikeskeinen menetelmä, roolin elinkaaren perustuva menetelmä, skenaarioperusteinen menetelmä, tavoiteperusteinen menetelmä ja integroitu menetelmä.

Arvioin tutkimuksen kohteena olevia menetelmiä käyttäen perustana Coynen esittämää ajatusta siitä mitä onnistuneen roolien määrittelyprosessin tuloksena tulisi syntyä. Näitä olivat roolit, käyttöoikeudet, roolihierarkia ja rajoitteet. Lisäsin Coynen esittämään listaan roolien etsimissuunnan ja soveltuvuuden erikokoisille yrityksille tai organisaatiolle. Tällä pyrin tuomaan paremmin esiin eri menetelmien käytännön hyötyjä.

Tutkimistani menetelmistä käyttökelpoisimmat olivat skenaarioperusteinen menetelmä ja eri matemaattisia menetelmiä runsaasti hyödyntävä integroiva menetelmä. Ne lähestyivät roolien määrittelyn ongelmaa hyvin erilaisista suunnista. Skenaarioperusteinen roolien määrittely hyödynsi vaatimusmäärittelyn kentällä paljon käytettyä, testattua ja toimivaksi todettua skenaarion käsitettä, ollen näin melko konservatiivinen lähestymistapa. Integroitu menetelmä edusti puolestaan liberaalimpaa ajatusmaailmaa. Menetelmä perustui organisaatiosta saatavan tiedon tilastolliseen käsittelyyn ja erilaisten louhinta-algoritmien hyödyntämiseen. Integroivan mallin käyttöä rajoittanee kaupallisten sovellusten saatavuus, ja menetelmän uutuudesta johtuva tutkimusnäytön puute.

Tutkielmani vahvuudeksi näkisin teoriaosan ja systemaattisen kirjallisuuskatsauksen toisiaan tukevan rakenteen. Teoriaosassa käsittelin roolipohjaisen pääsynhallinnan eri osa-alueita ja käsitteitä tarvittavalla tarkkuudella, jotta roolien määrittelyyn sisältyvä problematiikka olisi helpommin ymmärrettävissä roolien määrittelymenetelmien kuvauksissa ja vertailussa. Systemaattisen kirjallisuuskatsauksen huolellisella suunnittelulla ja eri vaiheiden tarkalla raportoinnilla pyrin prosessin läpinäkyvyyteen ja toistettavuuteen. Tämä lisää osaltaan tutkimukseni luotettavuutta.

Tutkimukseni tarjoama tieto eri roolien määrittelymenetelmien vahvuuksista tarjoaa peruspuitteet sopivan menetelmän valintaan erityyppisille yrityksille ja organisaatioille.

Näkisin lisätutkimustarvetta etenkin integroivan menetelmän toiminnan arvioimisessa. Käytännön hyötyä lisäisi myös roolien määrittelyyn erikoistuneiden sovellusten ominaisuuksien arviointi.

Sopivan roolien määrittelymenetelmän valinnalla voidaan vaikuttaa suotuisasti roolien määrittelyprosessin lopputulokseen. On kuitenkin huomattavaa, että tällöinkin tarvitaan eri alojen asiantuntijoiden yhteistyötä. Määrittelyprosessissa tarvitaan liiketoimintaprosessien syvällistä ymmärtämistä, eri järjestelmien teknisten yksityiskohtien ymmärtämistä ja tulevan järjestelmän käyttäjien tietämystä. Edellä mainittujen seikkojen huomioiminen kaikilta osiltaan varmistaa parhaan lopputuloksen.

LÄHTEET

[AlS04] J. Albanese ja W. Sonnenreich*: Network Security Illustrated*. McGraw-Hill, 2004, Yhdysvallat.

[Ans04] ANSI INCITS 359-2004: *Role Based Access Control*, American National Standard for Information Technology, 2004, Yhdysvallat.

[CoD07] E. Coyne ja J. Davis: *Role Engineering for Enterprise Security Management*. Artech House, 2007, Yhdysvallat.

[CoW08] E. Coyne ja T. Weil: An RBAC Implementation and interoperability Standard: The INCITS Cyber Security 1.1 Model, *IEEE Security & Privacy*, 2008, Yhdysvallat.

[Coy95] E. Coyne: Role engineering. *Proceedings of the first ACM Workshop on Role-based access control*, 1995, no. 4 (marraskuu 1995), 15-16.

[FeH97] E. Fernandez, J. Hawkins: Determining role rights from use cases. *Proceedings of the second ACM workshop on Role-based access control*, Fairfax, 1997, 121-125.

[FeK92] D. Ferraiolo ja D. Kuhn: Role Based Access Control. *15th National Computer Security Conference*, Baltimore, 1992, 554-563.

[FKC07] D. Ferraiolo ja D. Kuhn ja R. Chandramouli: *Role Based Access Control*. Artech House, 2007, Yhdysvallat.

[GGM10] C. Giblin ja M. Graf ja G. Karjoth ja A. Wespi ja I. Molloy ja J. Lobo ja S. Calo: Towards an Integrated Approach to Role Engineering. *Proceedings of the 3rd ACM workshop on Assurable and usable security configuration*, Chicago, 2010, 63-70.

[HeA03] Q. He ja A. Anton: *A Framework for Modeling Privacy Requirements in Role Engineering*, North Carolina State University, 2003, Yhdysvallat.

[HRU76] M. Harrison ja W. Ruzzo ja J. Ullman: Protection in Operating Systems. *CACM 19*, no. 8 (elokuu 1976), 461-471.

[KKS02] A. Kern ja M. Kuhlman ja A. Schaad ja J. Moffett: Observations on the Role Life-Cycle in the Context of Enterprise Security Management. *Proceedings of the 7th ACM symposium on Access control models and technologies*, Monterey, 2002, 43-51.

[NeS02] G. Neumann ja M. Strembeck: A Scenario-driven Role Engineering Process for Functional RBAC Roles. *Proceedings of the 7th ACM symposium on Access control models and technologies*, Monterey, 2002, 33-42.

[PCN04] J. Park ja K. Costello ja T. Neven ja J. Diosomito: A composite rbac approach for large, complex organizations. *Proceedings of the ninth ACM symposium on Access control models and technologies*, no. 9 (kesäkuu 2004), 163-172.

[RSW00] H. Roeckle ja G. Schimpf ja R. Weidinger: Process-oriented approach for role-finding to implement role-based security administration in a large industrial organization. *Proceedings of the fifth ACM workshop on Role-based access control*, no. 5 (kesäkuu 2000), 103-110.

[SaS94] R. Sandhu ja P. Samarati: Access Control: Principles and Practice. *IEEE Communications Magazine 32*, no. 9 (syyskuu 1994), 40-48.

[SCF96] R. Sandhu ja E. Coyne ja H. Feinstein ja C. Youman: Role-Based Access Control Models, *IEEE Computer 29*, no. 2 (helmikuu 1996), 38-47.

[Sul11] C. Sullivan: *Digital Identity: An Emergent Legal Concept*. University of Adelaide Press, 2011, Australia.

[Tie04] Tietotekniikan liitto: *Tietotekniikan liiton ATK-sanakirja*. Talentum, 2004, Suomi.

[TOB98] D. Thomsen ja D. O'Brien ja J. Bogle: Role Based Access Control for Network Enterprises. *Proceedings of the 14th Annual Computer Security Applications Conference*, Scottsdale, 1998, 50-58.

[Vir96] V. Gligor: Characteristics of Role-Based Access Control. *RBAC '95: Proceedings of the first ACM Workshop on Role-based access control*, no. 10 (joulukuu 1996), 9-14.

LIITE 1: Systemaattisen kirjallisuuskatsauksen hakutulokset

Tietokanta	Hakusanat ja hakutyyppi	Tulokset	Hyväksyttyjä
ACM	Haku=”rbac”	941	-
	Haku=”role engineer?”	9706	-
	Sanahaku=(role engineering process) Ja Sanahaku=(rbac)	477	6 (2 viitteistä)
IEEE Xplore	Sanahaku=(role engineering process) Ja Sanahaku=(rbac)	39	1 (1 viitteistä)
ScienceDirect	Sanahaku=(role engineering process) Ja Sanahaku=(rbac)	270	0
Web of Science	Sanahaku=(role engineering) Ja Sanahaku=(rbac)	17	ei uusia

Tietokanta	Artikkelin nimi
CiteSeerX	A Framework for Modeling Privacy Requirements in Role Engineering

LIITE 2: Systemaattisen kirjallisuuskatsauksen artikkelit

Tekijä	Vuosi	Artikkeli	Julkaisufoorumi
E. Coyne [Coy95]	1995	Role Engineering	ACM
E. Fernandez, J. Hawkins [FeH97]	1997	Determining Role Rights from Use Cases	ACM
D. Thomsen, D. O'Brien, J. Bogle [TOB98]	1998	Role Based Access Control for Network Enterprises	IEEE Xplore
H. Roeckle, G. Schimpf ja kumpp. [RSW00]	2000	Process-Oriented Approach for Role-Finding to Implement Role-Based Security Administration in a Large Industrial Organization	ACM
A. Kern, M. Kuhlman ja kumpp. [KKS02]	2002	Observations on the Role Life-Cycle in the Context of Enterprise Security Management	ACM
G. Neumann, M. Strembeck [NeS02]	2002	A Scenario-driven Role Engineering Process for Functional RBAC Roles	ACM
Q. He, A. Anton [HeA03]	2003	A Framework for Modeling Privacy Requirements in Role Engineering	CiteSeerX
C. Giblin, M. Graf ja kumpp. [GGM10]	2010	Towards an Integrated Approach to Role Engineering	ACM