



TAMPEREEN TEKNILLINEN YLIOPISTO

**MIKKO VÄLIMÄKI**

**XooNlps-sisällönhallintasovelluksen evaluointi tutkimus-  
tiedon tietokannaksi**

Diplomityö

Tarkastajat: professori Ulla  
Ruotsalainen, yliassistentti Jari  
Peltonen

Tarkastajat ja aihe hyväksytty  
Sähkö- ja tietotekniikan tiede-  
kuntaneuvoston  
kokouksessa 06.10.2010

# TIIVISTELMÄ

TAMPEREEN TEKNILLINEN YLIOPISTO

Tietotekniikan koulutusohjelma

**MIKKO VÄLIMÄKI: XooNlps-sisällönhallintasovelluksen evaluointi tutkimustiedon tietokannaksi**

Diplomityö, 57 sivua

Helmikuu 2012

Pääaine: Ohjelmistotuotanto

Tarkastajat: professori Ulla Ruotsalainen, yliassistentti Jari Peltonen

Avainsanat: tietokanta, käsitteellinen mallinnus, neurotiede, neuroinformatiikka, XooNlps

Neurotieteiden alalla on tarvetta tietokannoille tutkimustiedon monimuotoisuuden ja sen hallinnan vaikeuden vuoksi. Aivoja tutkivaa tietoa on haastavaa yhdistää ymmärrettäväksi kokonaisuudeksi, mikä on luonut tarpeen tietokantojen käytölle tutkimuksissa. Tietoa ei ole pystytty tallentamaan organisoidusti yhteen paikkaan.

Tässä työssä tutkimusryhmän tiedon hallintaan valittiin japanilaisten neuroinformaatikkojen kehittämä XooNlps-sisällönhallintasovellus. XooNlps:n soveltuvuutta tutkimusryhmän käyttöön arvioitiin ryhmältä kerättyjä vaatimuksia vasten. Arviointi sisälsi sovelluksen käyttökokeilujen lisäksi XooNlps:n tietokannan takaisinmallinnuksen käsitteelliseksi malliksi ja sen vertailun kerättyjen vaatimusten mukaan laadittuun uuteen malliin.

Vertailun tuloksena löydettiin yhtäläisyyksiä mallien välillä mutta myös eroja, joita uudelleen suunnitellulla mallilla pyrittiin löytämään. Uudelleen suunniteltu malli ottaa paremmin huomioon tietotyyppien metatietojen tallennuksen. Lisäksi tiettyjen tiedostotyyppien vaatimien otsikkotiedostojen tallennus onnistuu uudella mallilla. Vertailussa tuli esiin myös, että uudelleen suunnitellussa mallissa ei huomioida mahdollista kehittämiskohtaa hakemistopohjaisen tiedon tallennukseen, joka molemmissa malleissa on melko samankaltainen.

Vertailulla onnistuttiin todentamaan, että uudelleen suunniteltu malli paransi alkuperäistä mallia. Vertailussa tuli kuitenkin myös ilmi, että uudelleen suunnitellussa mallissakin on kehitystarvetta.

## ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY

Master's Degree Programme in Information Technology

**MIKKO VÄLIMÄKI : The evaluation of the XooNIps content management system for research data database.**

Master of Science Thesis, 57 pages

February 2012

Major: Software Engineering

Examiner: professor Ulla Ruotsalainen, senior assistant Jari Peltonen

Keywords: database, conceptual modelling, neuroscience, neuroinformatics, XooNIps

There is a need for databases in the field of neuroscience because of the diversity of its research data and the complexity of the data management. It is challenging to integrate the brain research data to one comprehensible entity, which has created a need for utilization of databases in research. It has not been possible to store the data organized in one place.

In this thesis, the XooNIps content management system was chosen for the research group's data management. The feasibility of XooNIps for research data management was evaluated with respect to the requirements gathered from the research group. The evaluation consisted of the testing the software's functionality, the reverse-engineering the XooNIps' database and comparing its model to the new designed model based on the original XooNIps model and the gathered requirements.

The result of the comparison showed some differences in the models. The designed model improves the metadata storing and also makes it possible to store header files used in some data types. It was also found that the directory based data storing was ignored in the designing of the new model since the storing is quite similar in both models.

The comparison between the models successfully shows that the designed model improved the original one. However it also became evident with the comparison, that there is some room for development in the designed model too.

## ALKUSANAT

Työ tehtiin Tampereen teknillisen yliopiston Signaalinkäsittelyn laitoksella M<sup>2</sup>oBSI-tutkimusryhmässä. Työn tarkastajana toimi professori Ulla Ruotsalainen Signaalinkäsittelyn laitokselta ja ohjaajana yliasistentti Jari Peltonen Ohjelmistotekniikan laitokselta. Ullaa ja Jaria haluan kiittää hyvistä neuvoista ja asiantuntevista kommentteista työn ohjauksessa. Lisäksi haluan kiittää työn alussa ohjausta antanutta Antti Pettistä sekä muita työssä auttaneita henkilöitä Signaalinkäsittelyn laitoksella.

Erityiskiitokset haluan antaa vanhemmilleni sekä kummisedälleni Antille, jotka ovat auttaneet ja tukeneet minua opintojeni alusta lähtien. Osoitan kiitokset myös vaimolleni Heidille, jonka kannustuksella sain tämän työn ja opinnot vihdoin päätökseen.

# SISÄLLYS

1. Johdanto . . . . .	1
1.1 Neurotieteet ja neuroinformatiikka . . . . .	1
1.1.1 Neuroinformatiikan historiaa . . . . .	1
1.1.2 Yleinen tietotekniikan tarve neurotieteissä . . . . .	2
1.1.3 Neurotieteissä käsiteltävä tieto . . . . .	4
1.1.4 Tietokantojen käyttö neurotieteiden ja bioinforma- tiikan aloilla . . . . .	6
1.2 Työn tausta ja rakenne . . . . .	8
2. Tietokantojen teoria ja taustatiedot . . . . .	10
2.1 Tietokannat . . . . .	10
2.1.1 Tietokannan hallintajärjestelmä . . . . .	10
2.1.2 Relaatiotietokanta . . . . .	13
2.2 Tietokantojen suunnittelu . . . . .	16
2.2.1 Vaatimusten keruu . . . . .	16
2.2.2 Käsitteellinen mallinnus . . . . .	17
2.2.3 Looginen suunnittelu . . . . .	18
2.2.4 Fyysinen suunnittelu . . . . .	18
2.2.5 Takaisinmallinnus . . . . .	19
2.3 Käsitekaavio . . . . .	19
2.4 Käytetyt tukisovellukset . . . . .	21
2.4.1 MySQL . . . . .	21
2.4.2 XOOPS . . . . .	22
2.4.3 XooNIps . . . . .	23
3. Tietokannan tarve tutkimusryhmässä . . . . .	25
3.1 Tietokannan tarve Signaalinkäsittelyn laitoksella . . . . .	25
3.1.1 M <sup>2</sup> oBSI-ryhmä . . . . .	25
3.1.2 Tutkimustieto tutkimusryhmässä . . . . .	26
3.1.3 Tietokannan vaatimukset . . . . .	27
3.1.4 Käyttötapaukset . . . . .	28
4. XooNIps . . . . .	29

4.1	XooNIps-sovelluksen esittely . . . . .	29
4.2	XooNIps:n käyttöönotto ja muokkaus . . . . .	30
4.3	XooNIps:n tietokanta . . . . .	31
4.4	XooNIps:n tietokannan takaisinmallinnus käsitteelliseksi malliksi . . . . .	32
5.	XooNIps:n evaluointi . . . . .	35
5.1	Käyttökokeilut XooNIps:lla . . . . .	35
5.2	XooNIps:n arviointi . . . . .	38
5.3	Tietokannan uudelleen suunniteltu malli . . . . .	38
5.3.1	Käsittekaavio . . . . .	39
5.3.2	Saantipolkuanalyysi . . . . .	42
5.3.3	Looginen suunnittelu . . . . .	43
5.4	XooNIps:in tietokannan ja uudelleen suunnitellun mallin vertailu . . . . .	44
5.4.1	Käsitteellisten mallien yhtäläisyydet . . . . .	44
5.4.2	Käsitteellisten mallien erot . . . . .	45
6.	Johtopäätökset ja työn arviointi . . . . .	47
6.1	Johtopäätökset . . . . .	47
6.2	Työn arviointi . . . . .	48
6.3	Vaihtoehtoja XooNIps:lle . . . . .	50
7.	Yhteenvedo . . . . .	51
	Lähteet . . . . .	53

## LYHENTEET JA TERMIT

Alaluokka	Luokka, joka on erikoistettu yläluokasta.
ANSI/SPARC	American National Standards Institute, Standards Planning And Requirements Committee.
DBMS	Database management system, tietokannan hallintajärjestelmä.
INCF	International Neuroinformatics Coordination Facility.
InnoDB	Transaktioita ja vierasavaimia tukeva tietokantamoottori MySQL:ssä.
Index	XooNips:n hakemisto, jossa voi olla Itemejä ja muita Indexejä.
Item	XooNips:ssa tietotyyppiä kuvaava käsite.
M <sup>2</sup> oBSI	Methods and Models for Biological Signals and Images, tutkimusryhmä TTY:n Signaalinkäsittelyn laitoksella.
MyISAM	Oletustietokantamoottori MySQL:ssä.
MySQL	Suosittu avoimen lähdekoodin tietokannanhallintajärjestelmä.
PHP	Hypertext Preprocessor, skriptikieli web-ohjelmointiin.
PET	Positroniemissiotomografia.
Portaali	Verkkopalvelu, joka kokoaa samantyyppisiä palveluita yhteen ja tarjoaa niihin pääsyn.
SQL	Structured Query Language, tietokantakieli.
TIFF	Tagged Image File Format, yleinen kuvien tallennukseen käytetty tiedostotyyppi.
Transaktio	Sarjana suoritettu joukko operaatioita, joka peruutetaan kokonaan, jos yksikin operaatio epäonnistuu.

Tutkimustieto	Tutkimuksessa käytettävä sekä tutkimuksen tuloksena syntyvä tieto.
UML	Unified Modeling Language, graafinen mallinuskieki.
XOOPS	Avoimen lähdekoodin web-sovellusalusta ja sisällönhallintasovellus.
XooNIps	XOOPS:n päälle neurotiedetutkimukseen kehitetty tiedon säilömiseen ja organisointiin tarkoitettu moduuli.
Yläluokka	Luokka, josta erikoistetaan alaluokkia.



# 1. JOHDANTO

## 1.1 Neurotieteet ja neuroinformatiikka

Neurotieteet ovat joukko tieteenaloja, joissa tutkitaan aivojen ja hermoston toimintaa ja rakennetta. Neurotieteet yhdistävät niin fysiikan, kemian ja biologian kuin fysiologian ja käyttäytymistieteiden aloja. Kyse on siis hyvin laajasta tieteenalojen yhdistelmästä, jonka perimmäinen tavoite on saada ymmärrys ihmisen aivojen ja hermoston toiminnasta sekä tämän suhteesta ihmisen käyttäytymiseen ja oppimiseen. Neurotieteissä tutkitaan esimerkiksi aivojen verenkiertoa erilaisien tunnetilojen ajan tai mitä aivoissa tapahtuu eri aistihavaintojen aikana.

Neuroinformatiikka on yksi tieteenala neurotieteissä. Neuroinformatikka keskittyy etsimään tapoja hyödyntää tietotekniikkaa neurotieteiden tutkimuksessa [1]. Siinä kehitetään työkaluja ja menetelmiä tutkimuksen edistämiseen ja tutkimustulosten hallintaan. Neuroinformatiikka on tulevaisuudessa ratkaisevasti mukana neurotieteiden päämäärän saavuttamisessa, sillä jatkuvasti kasvavan tietomäärän jäsennellylle tallennukselle on tarvetta. Tällä saadaan jaettua suuri määrä tietoa loogisesti, jolloin sen käyttäminen neurotieteiden tutkimuksessa on mahdollista. [2]

### 1.1.1 Neuroinformatiikan historiaa

Neurotieteet kehittyivät kovaa vauhtia 80- ja 90-luvuilla. Tietotekniikan kehitys ja yleistyminen mahdollistivat kehittyneempien tutkimusmenetelmien käytön aivojen tutkimisessa. Myös molekyylibiologian alan kehitys auttoi neurotieteitä saamaan parempaa ymmärrys-

tä aivoista ja sen toiminnasta, sillä aivotutkimuksessa hyödynnetään molekyylibiologiaa.[1] Ala onkin ollut tärkeä osa neurotieteiden ja neuroinformatiikan kehityksestä. Se alkoi siirtyä jo lähes kolme vuosikymmentä sitten tutkimuksessaan tietoteknisten työkalujen sekä verkossa olevien tietokantojen ja tietovarastojen käyttöön. Bioinformatiikka syntyi tämän kehityksen myötä palvelemaan molekyylibiologian tarpeita käyttäen tietotekniikkaa apuna tutkimuksessa.[3]

Neuroinformatiikka on edennyt samansuuntaisesti kuin bioinformatiikka, tosin reilun vuosikymmenen jäljessä.[3] Neuroinformatiikan sanotaan syntyneen vuonna 1993, kun Human Brain Project (HBP) [4] käynnistettiin edistämään tietotekniikan käyttöä neurotieteissä. Vuonna 1996 pidetyssä OECD-maiden (Organization for Economic Cooperation and Development) Global Science Forum -tapahtumassa (silloinen Mega Science Forum) Yhdysvaltain aloitteesta päätettiin aloittaa kansainvälinen yhteistyö vahvistamaan neuroinformatiikan panosta neurotieteiden tutkimuksessa.[1]

Tämä kansainvälinen yhteistyö sai konkreettiset kasvot vuonna 2005, kun vuotuisessa Global Science Forumissa perustettiin virallinen organisaatio neuroinformatiikalle, INCF (International Neuroinformatics Coordinating Facility) [5]. INCF:ään kuuluu 14 jäsenmaata, joilla kullakin on kansallinen yksikkö koordinoimassa neuroinformatiikan tutkimusta omassa maassaan. Perustamisestaan lähtien INCF on järjestänyt työryhmiä eri ongelmien ratkaisuun sekä laajempia ohjelmia pidemmän tähtäimen tavoitteisiin. Lisäksi vuodesta 2008 lähtien INCF alkoi järjestää Neuroinformatics-konferenssia, jossa neuroinformatiikan tutkijat kokoontuvat vuosittain tapaamaan toisiaan ja esittelemään tutkimustuloksiaan.[6]

### **1.1.2 Yleinen tietotekniikan tarve neurotieteissä**

Neuroinformatiikalle alkoi ilmetä tarvetta, kun tietämys aivojen toiminnasta kasvoi merkittävästi kahden viime vuosikymmenen aikana.[1]

Tutkimustietoa alkoi kertyä paljon, ja se oli saatava talteen. Ongelmaksi neurotieteissä on aina noussut tiedon monimuotoisuus. Tilanetta voidaan verrata neuroinformatiikan tiennäyttäjään, bioinformatiikkaan. Siinä käsiteltävä tutkimusaineisto on lähinnä samanmuotoista sekvenssitietoa. Tällaista yhdenmukaista tietoa on helpompi hallita ja käsitellä. Neurotieteissä tutkimusta tehdään eri menetelmillä ja biologisesti monilla tasoilla. Aivoja voidaan tutkia esimerkiksi solu- ja molekyyli-tasolla tai vaikka koko aivot kattavalla aivokuvannuksella. [2]

Tutkinnan kohteita on siis paljon ja näitä tutkitaan eri menetelmillä ja laitteistoilla. Tämän vuoksi neurotieteissä tiedon hallinta on haastavaa, koska tutkimusaineisto on monimuotoista ja sitä on paljon. Tietoa on vaikea yhdistää yhdeksi kokonaisuudeksi [1]. Sitä joudutaan tallentamaan useissa eri formaateissa, eri tavoilla ja mahdollisten kapasiteettirajoitusten takia eri tallennusmedioihin. Tämä tuo entistä enemmän vaikeuksia tiedon yhdistämiseen. Tähän kuitenkin pyritään, koska ymmärrystä koko aivojen ja hermoston toiminnasta ei voida saavuttaa ymmärtämällä vain osakokonaisuuksia, vaan siihen vaaditaan laajempi näkemys.

Haasteeseen tietotekniikan tarpeesta on vastattukin jo. Esimerkiksi Norjassa toimiva NeSys (Neural Systems and Graphics Computing Laboratory) [7] on neuroanatomian ja neuroinformatiikan laboratorio, joka tekee tietokantaratkaisuja neuroinformatiikassa muun muassa automaattisen kuvatiedon hankinnan, 3D-kuvien uudelleenrakennuksen ja tiedon hallinnan suhteen. Laboratorio on osa Oslon yliopiston molekyylibiologian ja neurotieteiden keskusta (Centre for Molecular Biology and Neuroscience) [8], jossa yhdistyy molekyylibiologia, bioinformatiikka ja neurotieteet aivosairauksien tutkinnassa.

### 1.1.3 Neurotieteissä käsiteltävä tieto

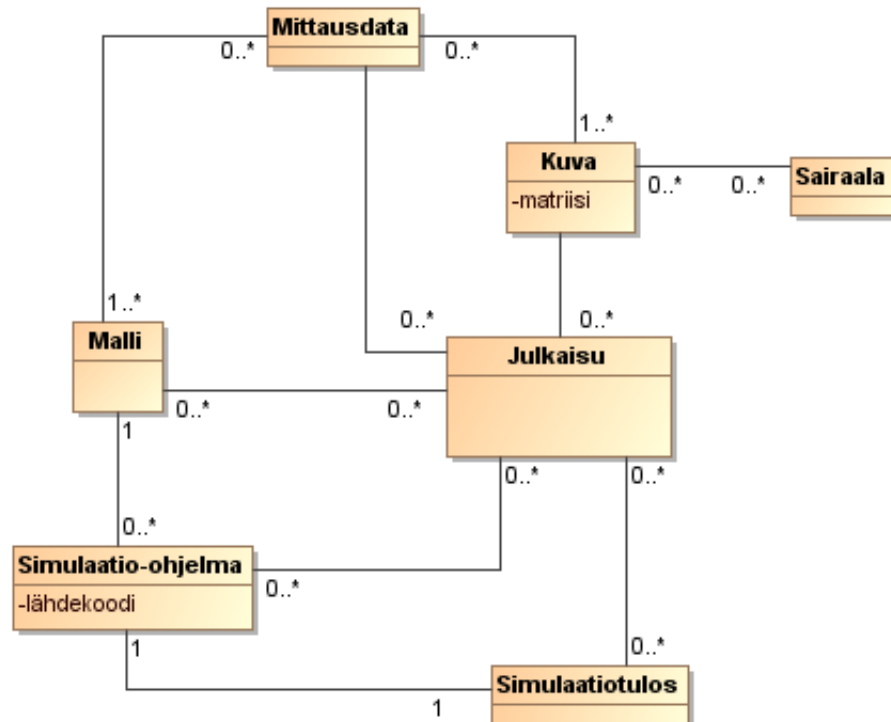
Neurotieteissä tutkimustietoa on monenlaista. Se voi olla esimerkiksi hyvin yksityiskohtaista molekyyli-tason tietoa geeneistä tai soluista, tai tietoa synapsien rakenteesta eri tasoilla hermostorakennetta. Tutkimustietoa on myös aivosähkökäyrätieto, aisti- ja tietoisten toimintojen havainnointitieto tai aivokuvannuksen tulokset, jotka voidaan jakaa vielä mittaustietoon ja niistä luotuihin kuviin. Kuvia on kaksi-, kolmi- sekä neliulotteisia (aikataso). Kuvatiedosta voidaan generoida myös videota, jos kuvia on otettu aikatasossa. [9]

Mittauksista saatavan tiedon tiedostomuoto voi vaihdella. Tietoa saa useiten numeerisena, jolloin sitä on mahdollista käyttää helposti, esimerkiksi matriisimuodossa olevia kuvia voi käsitellä useilla ohjelmistoilla. Joidenkin mittalaitteiden antama tulostieto on vain tälle mittalaitteelle suunniteltujen ohjelmistojen ymmärtämiä tiedostomuotoja. Tällaista tietoa on vaikeampi käyttää, jos ohjelmistoja ei ole tarjolla. Neurotieteissä tiedon määrä on erittäin suuri, käytännössä rajaton, koska kuvatieto on useampiulotteista ja kuvia saadaan lukemattomista kohteista.

### Kuvatiedon käyttö neurotieteissä

Kuvassa 1.1 esitetään yleinen malli kuvatiedon käytöstä ja siihen liittyvistä käsitteistä, joita neurotieteiden tutkimuksessa voi esiintyä.

Mittauksilla tutkitaan potilasta eri menetelmillä. Mittauksista saadaan mittaustietoa, josta voidaan generoida kuva ja tehdä yleinen malli mitatun tapahtuman tai ilmiön esittämiseksi. Kuva jää sairaalaan ja se voidaan antaa eteenpäin tutkimuskäyttöön. Kun se annetaan tutkimuskäyttöön, kuva anonymisoidaan, eli siitä poistetaan kaikki tiedot, joilla kuva voitaisiin yhdistää potilaan henkilöllisyyteen [10]. Tämän vuoksi kuvatietoa ei voi säilyttää esimerkiksi potilastietojen mukaan jaoteltuna.



Kuva 1.1: *Kuvatiedon käyttö neurotieteissä*

Malleja käytetään simuloimaan tutkimuskohteiden toimintaa, kuten esimerkiksi veren virtausta aivojen aktiivisessa osassa. Malleista voidaan tehdä simulaatio-ohjelmia, joilla ajetaan simulaatioita. Simulaatioiden tulosten avulla voidaan analysoida mitattua tapahtumaa tai ilmiötä. Niiden perusteella voidaan myös analysoida mallia mahdollisia muutoksia varten. Kaikki tämä tutkimuksessa käytetty tieto ovat lopulta osa julkaisua, jossa esitellään tutkimuksen tulokset.

Tutkimustyössä syntynyttä tietoa (mallit, simulaatio-ohjelmat ja -tulokset) voidaan käyttää myös muissa tutkimuksissa ja julkaisuissa. Kun samat tutkijat ja sama tutkimusryhmä tekevät töitä vain muutama vuoden ajan, tuloksena voi olla useampikin julkaisu. Tämä hankaloittaa tiedonhallintaa, koska tietoa on paljon eikä sitä hallita organisoidusti. Kun uuteen tutkimukseen tarvitaan vanhoja tutkimustuloksia, näitä ei välttämättä löydy. Jos tarvitaan esimerkiksi simulaatioiden tuloksia, päiviäkin kestävät simulaatiot voidaan joutua pahimmassa tapauksessa ajamaan kokonaan uudestaan.

## Positroniemissiotomografia, PET

Positroniemissiotomografiassa (PET) tutkitaan potilaan elimistön biologisia prosesseja. Tässä menetelmässä potilaan elimistöön injektoidaan merkkiaine, johon on sekoitettu radioaktiivista ainetta. PET-kamera havaitsee nyt radioaktiivisen merkkiaineen käyttäytymisen. Tällä tavalla voidaan seurata merkkiaineen kulkua elimistössä. Valitsemalla tietty merkkiaine (johon radioaktiivinen aine liitetään), voidaan seurata haluttuja biologisia prosesseja elimistössä, kun tiedetään tämän merkkiaineen liikkeit. Aivojen tutkinnassa merkkiaine voi koostua esimerkiksi glukoosista, koska aivojen aktiivisissa osissa verenkierto ja sokerin käyttö kasvavat. Tällöin voidaan seurata biologisia prosesseja aivojen aktiivisissa osissa.[11, 12]

### 1.1.4 Tietokantojen käyttö neurotieteiden ja bioinformatiikan aloilla

Neurotieteiden alalla on tällä hetkellä käytössä useita tietokantoja. Näitä ovat esimerkiksi ModelDB [13], NeuronDB [13] ja NeuroMorpho [14]. ModelDB ja NeuronDB ovat osa SenseLab-projektia [15], joka on Human Brain Projectin rahoittama hanke neurotieteiden tutkimuksen tukemiseen. SenseLab käsittää yhteensä kahdeksan tietokantaa eri aihealueisiin jaoteltuna [16].

Mainituista tietokannoista ModelDB:ssä on tarjolla laskennallisen neurotieteen malleja. Mallit ovat pseudokoodina, joten ne kuvaavat mallin logiikan ja niitä voidaan siis käyttää minkä tahansa ohjelmointikielen ja ympäristön kanssa. NeuronDB kokoaa yhteen tietoa kolmesta eri neuronin ominaisuudesta (jänniteohjattu sähköjohtavuus, välittäjäaineen vastaanottaja, välittäjäaineen aines).

NeuroMorpho on National Institutes of Healthin rahoittama hanke, jonka tarkoitus on tarjota tietokanta digitaalisesti tallennetuista neuronien eri muodoista. Tietoja saadaan useista eri laboratorioista

ja eri eläinlajeista. Tutkimuksista on tallessa paljon metatietoja kuten koejärjestelyt, kokeen suorittajat ja tietoja kohteesta.

Neurotieteiden alalla on siis käytössä erilaisia tietokantoja. Yhteinen tekijä niissä on niiden aihealueiden yksityiskohtaisuus, sillä mitään yleistä tietokantaa kaikenkokoisen neurotieteiden tiedon säilömiseen ei ole. Näitä tietokantoja yhteenkokoava palvelu oli pystyssä useita vuosia. Brain Information Groupin kehittämä NDG (Neuroscience Database Gateway) oli portaali [17], joka piti kirjaa eri tietokannoista, joissa säilytetään neurotieteiden tutkimukseen liittyvää tietoa. Nykyään NDG on liitetty osaksi uudempaa portaalia, NIF:iä (Neuroscience Information Framework) [18, 19]. NIF tarjoaa keskitetyn hakupalvelun kaikenlaisen neurotieteiden tiedon ja työkaluohjelmien etsintään. Se hyödyntää eri tietokantoja ja aivoatlaksia, joista se hakee käyttäjän antamien hakusanojen mukaan tietoa. Haettava tieto tuodaan esille NIF:n käyttöliittymään, joka antaa hakutuloksiin linkit niiden alkuperäiseen lähteeseen. Mainittu NeuroMorpho on yksi NIF:n käyttämistä tietokannoista.

Myös bioinformatiikan alalla on käytössä monia eri tietokantoja. Esimerkiksi BioModels Database tarjoaa testattuja bioinformatiikan malleja, joita on linkitetty muihin resursseihin kuten julkaisuihin [20]. BioModels Databasessa itse malleja ei säilytetä tietokannassa, vaan tietokantaan tallennetaan ainoastaan otsikkotiedot malleista, itse mallit säilötään erikseen muualla [21].

Neurotieteiden alalle tällainen lähestymistapa tiedon säilytykseen on järkevä, koska tieto on heterogeenistä ja siten sen suuruusluokka vaihtelee paljon. Kun kuvatiedon suuruusluokka nousee useampaan gigatavuun, on sen tallennus tietokantaan resurssien tuhlaamista. Kuvan otsikkotiedoston tallennus tietokantaan ja itse kuvan tallennus muualle on silloin järkevä ratkaisu.

## 1.2 Työn tausta ja rakenne

Työ on ohjelmistotuotannon diplomityö ja se on tehty Tampereen teknillisen yliopiston Signaalinkäsittelyn laitoksella Methods and Models for Biological Signals and Images -tutkimusryhmässä (M<sup>2</sup>oBSI). Työ jakautuu kahteen osaan, käytännön osaan sekä teoreettiseen osaan. Käytännön osuus työssä on saada tutkimusryhmälle tietokanta tutkimustiedon säilytykseen. Käytettäväksi sovellukseksi oli jo valittu japanilaisten neuroinformatiikan tutkijoiden kehittämä XooNIps, joka on XOOPS-sisällönhallintasovelluksessa toimiva moduuli. XooNIps on kehitetty neuroinformatiikan tutkijoiden apuvälineeksi tutkimustiedon organisoituun hallintaan, joka saavutetaan ennalta määriteltyjen tietotyyppeiden avulla.

Tutkimusryhmän ongelma oli tutkimuksessa syntyvän tiedon hallinta. Tutkimuksessa käytetty tieto ja tutkimustulokset jäivät aina yksittäisen tutkijan koneelle, eikä kaikkien käytettäväksi. Tämä vaikeuttaa tutkimusta, koska tutkijat tarvitsevat myös muiden tutkimustuloksia omassa työssään. Ongelman ratkaisuksi keksittiin tietokanta, johon tutkijat tallentavat valmiita julkaisujaan ja niihin liittyvää tutkimustietoa kuten malleja, simulaatioita ja kuvia.

Työn teoreettisessa osuudessa tarkastellaan käytettävää XooNIps-sovellusta ja sen käyttämää tietokantaa. Sovelluksen tietokanta takaisinmallinnetaan käsitteelliseksi malliksi, jonka ominaisuuksia arvioidaan vaatimuksia vasten. Mallia pohjana käyttäen suunnitellaan uusi malli, jota verrataan alkuperäiseen. Vertailun tuloksia arvioidaan erikseen. Takaisinmallinnuksesta saatavan mallin arvioinnilla ja mallien vertailulla saadaan selville XooNIps:n tietokannan heikkoudet ja toisaalta uudelleen suunnitellun mallin suunnittelutyön tulokset.

Työ on ohjelmistotuotannon diplomityö, mutta se on tehty TTY:n Signaalinkäsittelyn laitokselle. Tämä asettaa tiettyjä vaatimuksia aiheen lähestymistapaan. Työssä käsitellään tietokannan suunnittelua käsitteiden osalta, mutta suunnittelussa ei syvennytä tietokannan tek-



nisiin yksityiskohtiin, koska sen ei katsota olevan oleellista työn asiayhteydessä.

Luvussa 2 esitellään työn ymmärtämiseen vaadittava teoreettinen osuus. Teoreettinen osuus muodostuu tietokantojen teoriasta, eli yleisesti mikä on tietokannan tarkoitus ja työn aiheeseen liittyvän relatiotietokannan rakenne sekä toimintaperiaate. Lisäksi luvussa käydään läpi tietokannan suunnitteluprosessi pääpiirteittäin ja esitellään työssä käytetyt tukisovellukset.

Luvussa 3 kerrotaan taustoja neurotieteiden alan tarpeesta tietokantojen käytölle. Esimerkkinä tästä käytetään tutkimusryhmän tarvetta tutkimustiedon säilytykselle. Luvussa esitellään käyttötapauksia XooNIps:n kanssa sekä tietokannan vaatimukset.

Luvussa 4 esitellään XooNIps-sovellus, joka otettiin käyttöön TTY:n Signaalinkäsittelyn laitoksen tutkimusryhmässä. XooNIps on sisällönhallintaan tarkoitettu web-sovellus, jonka käyttöönotto, muokkaus ja käyttökokeilujen tulokset esitellään. Pääasiassa luvussa tarkastellaan XooNIps:n tietokannan rakennetta.

Luvussa 5 arvioidaan XooNIps-sovelluksen tietokantaa ja kerrotaan käyttökokeiluista. Niiden pohjalta ja luvussa 3 esiteltyjä vaatimuksia vasten tietokantaa evaluoidaan ja tehdään uudelleen suunniteltu malli. Luvun lopussa alkuperäisen tietokannan käsitteellistä mallia ja uudelleen suunniteltua mallia vertaillaan.

Luvussa 6 esitetään johtopäätökset käsitteellisten mallien vertailusta. Luvussa myös arvioidaan työn tekemistä ja saatuja tuloksia. Lopuksi esitellään vaihtoehtoja XooNIps:lle.

## **2. TIETOKANTOJEN TEORIA JA TAUSTATIEDOT**

Tässä luvussa kerrotaan teoriaa tietokannoista sekä tietokannan suunnittelusta vaiheineen, käsitteellinen mallinnus ja looginen suunnittelu. Lisäksi luvussa esitellään työssä käytetyt teknologiat sekä tukisovellukset.

### **2.1 Tietokannat**

Tietokanta on kokoelma loogisesti toisiinsa liittyvää tietoa, jolla kuvataan rajattu kohdealue reaalimaailmasta. Se on säilytyspaikka järjestetylle tiedolle, jota useat käyttäjät voivat käyttää samanaikaisesti. Tietokanta on sähköisen tiedon tallennukseen käytetty väline sen nopeuden ja tiedon järjestelykyvyn vuoksi. [22]

#### **2.1.1 Tietokannan hallintajärjestelmä**

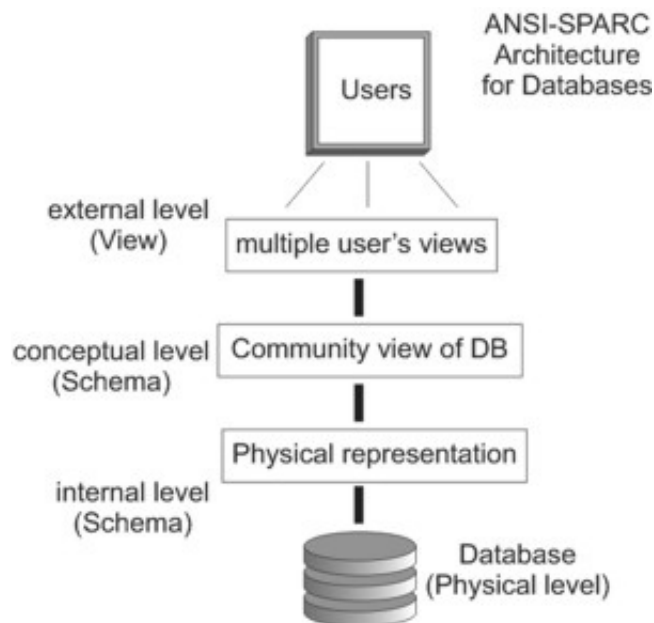
Tietokannan hallintajärjestelmä on ohjelma, joka määrittelee tavan tallentaa ja käsitellä tietoa tietokannassa. Tietokannan hallintajärjestelmää käytetään tietokantaa käyttävien sovellusten ja itse tietokannan välissä. Tietokannan käyttöoikeudet, samanaikainen käyttö, tiedon eheys ja virhetilanteet ovat tietokannan hallintajärjestelmän vastuulla.

Käyttöoikeuksilla rajoitetaan erilaisten käyttäjien pääsyä tietokannan sisältöön sekä oikeuksia tehdä kyselyjä tai muutoksia tietokantaan. Tietokannan samanaikainen käyttö vaatii transaktiota. Transaktio on joukko operaatioita, joiden suoritus onnistuu tai epäonnistuu kokonaisuutena. Tiedon eheydellä tarkoitetaan tietokannan tietojen

ristiriidattomuutta. Tietokannan hallintajärjestelmä suorittaa toipumisen virhetilanteista (ohjelmisto- tai laitteistovikojen aiheuttamat) palauttamalla tietokannan viimeisimpään virheettömyyden tilaan ja hoi-  
taa myös varmuuskopioinnin. [22]

## Tietokannan hallintajärjestelmän käsitteellinen arkkitehtuuri

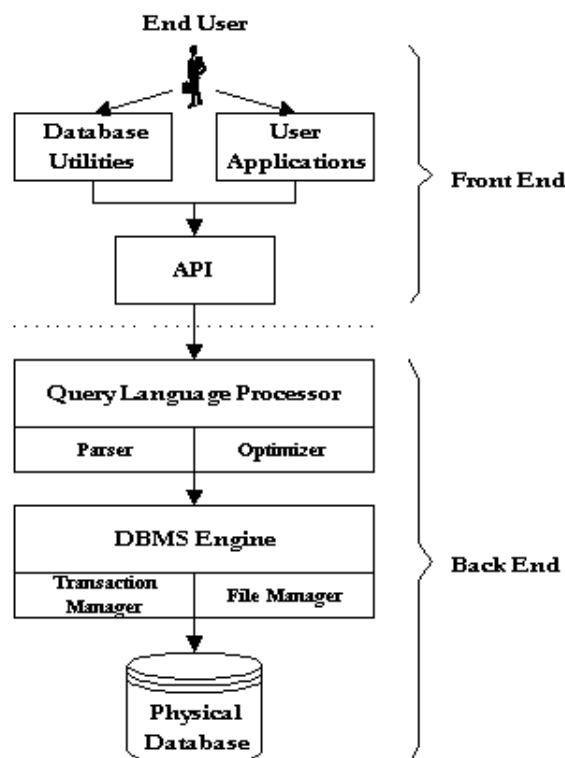
Tietokannan hallintajärjestelmä jaetaan useimmiten ANSI/SPARC-standardoituun arkkitehtuuriin perustuen käsitteellisesti kolmeen abstraktiotasoon, jotka on esitelty kuvassa 2.1. Ulkoinen taso (*external level*) näyttää tietokantaa käyttäville sovelluksille niille tarkoitetut näkymät tietokannasta. Se näyttää kaiken oleellisen ja sovelluksen tarvitseman tiedon, muttei mitään ylimääräistä. Käsitteellinen taso (*conceptual level*) määrittää mitä tietoa tietokantaan tallennetaan ja miten tiedot liittyvät toisiinsa. Sisäisellä tasolla (*internal level*) määritetään, miten tietokanta tallentaa tiedon fyysiselle levyille (*physical level*). Merkittävintä tietokannan hallintajärjestelmän käsitteellisessä arkkitehtuurissa on tasojen riippumattomuus toisistaan. [22]



Kuva 2.1: Tietokannan hallintajärjestelmän käsitteellinen arkkitehtuuri [22].

## Tietokannan hallintajärjestelmän fyysinen arkkitehtuuri

Tietokannan hallintajärjestelmän fyysinen arkkitehtuuri koostuu muutamista tärkeistä osista ja se on esitetty kuvassa 2.2. Edusta *front end* on käyttäjän ja sovellusten näkemä rajapinta tietokantaan. Tausta *back end* koostuu kyselynkäsittelijästä (*query language processor*), tietokantamoottorista (*DBMS engine*) sekä fyysisestä levystä (*physical database*), jonne tieto tallennetaan. Kyselynkäsittelijä ottaa vastaan käsittelypyyntöjä ylemmältä tasolta, jäsentää ja optimoi ne ja antaa tietokantamoottorille. Tietokantamoottorin tapahtumankäsittelijä (*transaction manager*) hoitaa operaatioiden suoritusjärjestyksen ja ajankohdat ja antaa tapahtuman yksilöivän tapahtumakahvan muistinhallitsijalle (*file manager*). Muistinhallitsija hoitaa tiedonsiirron fyysiselle levyille. Tietokantamoottoreita on erityyppisiä, ja tyyppistä riippuu muu muassa miten tietokannan samanaikainen käyttö, varmuuskopiointi ja virhetilanteista toipuminen toimivat. [22]



Kuva 2.2: Tietokannan hallintajärjestelmän fyysinen arkkitehtuuri [23].

## Tietomalli

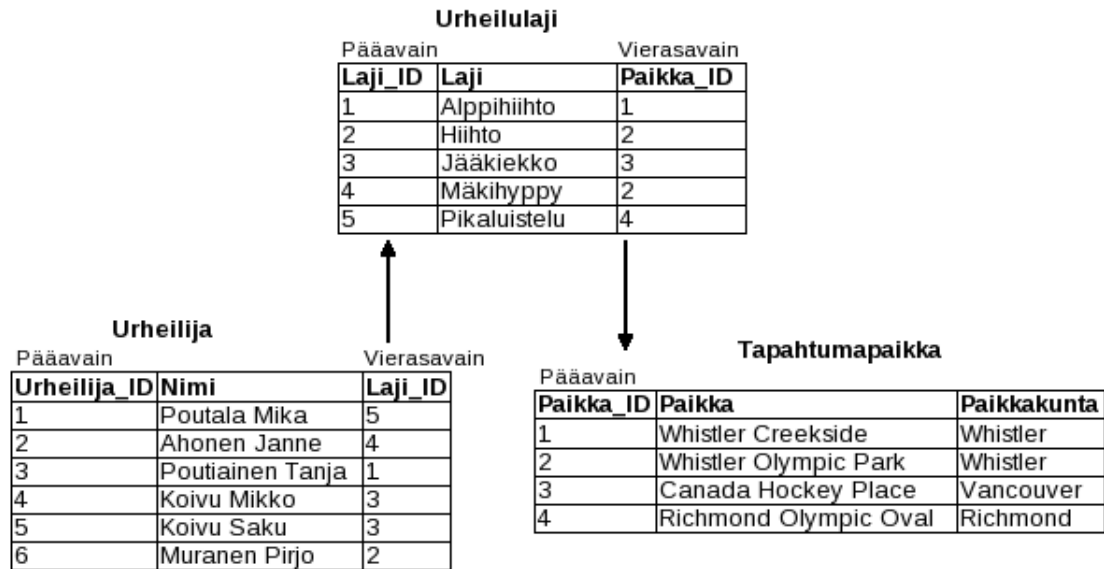
Tietokannan hallintajärjestelmä hallitsee ja käsittelee tietokannan tietoja valitun tietomallin perusteella. Tietomalli on esitystapa, jolla tietokannan kuvaama reaali maailman kohdealue esitetään. Sillä myös määritellään tapa käsitellä tietokannan tietosisältö. Tietomalleja on useita ja yksi yleinen tietomalli on relaatiomalli, jota käytetään myös tässä työssä. [22]

Muita tietokannoissa käytettyjä tietomalleja ovat muun muassa oliomalli, verkkomalli, hierarkkinen malli ja olio-relaatiomalli. Oliomallissa käsitteet kuvataan olioina, joilla on tietty tila, toiminta ja suhteet muihin olioihin. Verkkomalli on graafimainen verkko, jossa käsitteet ovat solmuja ja niiden väliset suhteet kaaria. Hierarkkisessa mallissa käsitteet ovat myös solmuja, mutta verkkomallista poiketen verkko on puugraafi, jossa solmulla voi olla vain yksi isä. Olio-relaatiomallissa on relaatiomalli, jossa on käytössä myös oliot. [22]

### 2.1.2 Relaatiotietokanta

Relaatiotietokannassa käytetään relaatiomallia, joka pohjautuu matemaattisiin relaatioihin. Relaatio koostuu ominaisuuksista ja monikoista, ja esittää jotakin reaali maailman käsitettä. Monikot ovat relaation esittämän käsitteen ilmentymiä, joilla on kaikilla relaatiolle määritellyt ominaisuudet. Relaatiomallin mukaisen tietokannan tiedot esitetään yleensä taulukkomuodossa, jossa relaatiota vastaa taulu. Taulussa on sarakkeita, jotka vastaavat relaation ominaisuuksia, sekä rivejä, jotka vastaavat relaation monikoita.

Kuvassa 2.3 on esimerkki taulukoiden käytöstä relaatiomallin esitystapana. Esimerkissä on kolme relaatiota esittämässä tietokantaa Vancouverin olympialaisten Suomen joukkueen urheilijoista (Urheilija-relaatio), heidän urheilulajeistaan (Urheilulaji-relaatio) sekä urheilulajien tapahtumapaikoista (Tapahtumapaikka-relaatio). [22]



Kuva 2.3: Esimerkki relaatiomallin taulukkomuotoisen esitystavan käytöstä.

Relaatiotietokannassa on käytössä rajoitteita, joilla rajataan relaatiomallin kuvaamaa kohdealuetta ja sen tietosisältöä. Pääavain on ominaisuus tai ominaisuuksien yhdistelmä, jolla yksilöidään jokainen monikko relaatiossa. Kuvassa 2.3 Urheilija-relaation pääavain on Urheilija\_ID, joka yksilöi jokaisen monikon eli urheilijan tässä relaatiossa. Pääavaimen arvo ei voi olla koskaan tyhjä arvo.

Vierasavain on viittaus relaation pääavaimen. Vierasavaimen ominaisuuksien arvot viittaavassa relaatiossa vastaavat viitatus avainehdokkaan ominaisuuksien arvoja jokaisen monikon kohdalla. Vierasavain voi viitata samaan relaatioon tai eri relaatioon. Kuvassa 2.3 relaatioiden vierasavaimet (Laji\_ID, Paikka\_ID) viittaavat eri relaatioihin. [22]

Tietosisältöä rajoitetaan myös ominaisuuksien arvoaluerajoitteilla. Relaation ominaisuuksille määritetään arvoalueet, jotka määräävät minkä tyyppistä tietoa kuhunkin ominaisuuteen tallennetaan. Kuvassa 2.3 Tapahtumapaikka-relaation Paikka-ominaisuuden arvoalueen rajoitteena voisivat olla esimerkiksi kaikki kanadalaiset paikannimet. [22]

Relaatiotietokannan luomiseen ja käyttöön tarvitaan tietokantakieli. Yleisin kieli relaatiotietokannoille on SQL [24]. Se on lyhenne sanois-

ta Structured Query Language, joka tarkoittaa rakenteellista kyselykieltä. SQL:llä tehdään kyselyjä tai muokkauskomentoja tietokantaan. Kyselyt tai komennot annetaan tietokannan hallintajärjestelmän kyselynkäsittelijälle, joka optimoi annetun käskyn ja palauttaa tuloksen. SQL:llä voidaan siis hakea, lisätä ja muokata tietokannan tietoa.

SQL:ssä peruskysely koostuu kolmesta osasta. *SELECT*-lauseella valitaan taulujen sarakkeet, jotka tulevat näkyviin tulokseen. *FROM*-lauseella valitaan taulut, joihin *SELECT*-lause kohdistetaan. *WHERE*-lauseella määritetään ehdot rivien arvojen tarkistukseen. Rivi otetaan mukaan kyselyn tulokseen jos sen arvot läpäisevät *WHERE*-lauseen ehdot. [24, 25]

## Relaatiotietokannan eheys

Relaatiotietokannan eheys tarkoittaa sen tietojen saavutettavuutta ja oikeellisuutta. Eheys voidaan jakaa kahteen lajiin, rakenteelliseen eheyteen ja semanttiseen eheyteen. Rakenteelliseen eheyteen kuuluu olioeheys sekä viite-eheys. Olioheys tarkoittaa, että taulun pääavain ei voi olla tyhjä arvo. Viite-eheys tarkoittaa, että vierasavaimen arvo tai arvojoukko viittaa aina olemassa oleviin arvoihin. [26]

Semanttinen eheys takaa tietokannan tietosisällön oikeellisuuden niiden sääntöjen ja rajoitusten puitteissa, mitä sille on määrätty. Relatioissa on vain niille asetettujen rajoitusten mukaista tietoa ja relaatioiden suhteet toisiinsa ovat määrättyt. [26] Esimerkiksi kuvassa 2.3 Urheilija-relaatiossa on vain suomalaisia talviurheilijoita ja Urheilulaji-relaatiossa on vain talviolympialaisten urheilulajeja. Lisäksi relaatioiden suhde määrää jokaiselle urheilijalle jonkin urheilulajin.

Kuvan 2.3 esimerkissä on käytetty myös vierasavaimia. Urheilija-relaatio viittaa Laji\_ID- vierasavaimella Urheilulaji-relaation Laji\_ID-pääavaimeen ja yhdistää näin urheilijat urheilulajeihinsa. Urheilulaji-relaatio viittaa Paikka\_ID-vierasavaimella Tapahtumapaikka-relaation

Paikka\_ID-pääavaimen ja yhdistää urheilulajit niiden tapahtumapaikkoihin.

Viite-ehyettä ylläpidetään asettamalla vierasavaimelle viittauspolitiikka. Viittauspolitiikka määrää, mitä vierasavaimen arvolle tai arvojoukolle tehdään, kun viitattua arvoa tai arvojoukkoa muutetaan tai se poistetaan. Viittauspolitiikka määritetään muutokselle ja poistolle erikseen. Viittauspolitiikkoja on erilaisia, kuten ketjupolitiikka, jossa vierasavaimen arvo seuraa viitattun avaimen arvoa. Estopolitiikassa esitetään vierasavaimen arvon muutokset tai poistot. Nolla-arvopolitiikassa arvoksi asetetaan nolla ja oletuspolitiikassa ominaisuudelle määritetty oletusarvo. Tässä työssä ei ole käytetty vierasavainpolitiikkoja. [22]

## **2.2 Tietokantojen suunnittelu**

Tietokannan suunnitteluun on pääasiassa kaksi erilaista lähestymistapaa. Toinen tavoista on kerätä yksityiskohtaista tietoa kohdealueen eri osista ja laajentaa kokonaiskuvaa askel kerrallaan. Tämä on vähemmän käytetty tapa, jota ei tarkastella tässä työssä. Yleisempi tapa on hahmottaa kohdealue kokonaisuutena ja siirtyä askel kerrallaan pienempiin palasiin. Tässä tavassa vaatimusten keruun jälkeen tehdään käsitteellinen mallinnus, looginen suunnittelu ja vielä fyysinen mallinnus ennen kuin tietokanta otetaan käyttöön. Tämä on pääasiallinen tapa tietokannan suunnittelussa, mutta vaiheiden suorituksen aloittamista ei ole tarvetta aloittaa täsmällisesti edellisen vaiheen loputtua, vaan vaiheet limittyvät keskenään. [22]

### **2.2.1 Vaatimusten keruu**

Ennen varsinaisen suunnittelun aloitusta on kerättävä vaatimukset tietokannalle. Vaatimusten keruu on ohjelmistotuotannon alalla tyypillinen suunnittelun esivaihe, jossa selvitetään mitä suunniteltavan ratkaisun on määrä tehdä. Tietokannan tapauksessa vaatimuksissa tulisi



olla selvitetty, mitä tietoa tietokantaan halutaan laittaa ja mitä tietoa sieltä halutaan hakea. Tietokannan suorituskykyä ja käyttöoikeuksia on myös hyvä miettiä jo tässä vaiheessa.

Esimerkkinä vaatimuksesta voisi olla, että tietokantaan halutaan tallentaa tiedot kaupan kaikista tuotteista ja tietokannasta voisi hakea eri tietoja tuotteista tuotteen numeron perusteella. Toinen vaatimus tähän tietokantaan voisi olla, että halutaan etsiä kaikki tuotteet, jossa on raaka-aineena maitoa. Vaatimusten keruulla yritetään siis saada selville reunaehdot tietokannan käyttötarkoitukselle. Kaikkia vaatimuksia ei ole kuitenkaan tarpeen eikä yleensä mahdollistakaan kerätä ennen käsitteellistä mallinnusta. [22]

### 2.2.2 Käsitteellinen mallinnus

Käsitteellistä mallinnusta tehdään vaatimusten keruun kanssa rinnakkain. Vaatimusten analysoinnin aikana mallinnetaan tallennettavan tiedon kohdealuetta, jotta oikeat vaatimukset saadaan selville. Vaatimusten pohjalta mallinnetaan järjestelmän käsitteellinen malli. Käsitteellinen malli on tietokannan kohdealueen kuvaus ja tietosisältö, ja se on riippumaton käytettävästä tietomallista tai tietokannan hallintajärjestelmästä. Käsitteellisessä mallissa kuvataan tietokannan kuvaama reaali maailman kohdealue käsitteillä sekä niiden ominaisuuksilla ja keskinäisillä suhteilla. Sen on tarkoitus kuvata koko kohdealue, kaikki sen tiedot, säännöt ja rajoitteet sekä myös sen eri tilat ja tilojen muutokset. Tämän vuoksi vaatimukset voivat vielä muuttua mallinnuksessa. [22]

Käsitteellinen malli koostuu tyypillisesti kahdesta osasta. Käsittekaaviossa kuvataan graafisesti kohdealueen käsitteet, niiden väliset suhteet sekä käsitteiden ominaisuudet. Tietohakemistossa määritellään käsitteiden ominaisuudet ja ominaisuuksien arvoalueet, sekä mahdolliset rajoitteet arvoalueisiin. [22]

## Saantipolkuanalyysi

Saantipolkuanalyysillä varmennetaan, että käsitteellisen mallin mukaisesta tietokannasta saadaan haettua kaikki tarvittava tieto. Analyysissä käydään läpi käsittekaaviosta reitit, joilla tarvittavia tietoja haetaan. Tällä tavoin löydetään virheet mallista, jos siitä ei saakaan tiettyjä tietoja tai tietojen haku on tarpeettoman hankalaa. [22]

### 2.2.3 Looginen suunnittelu

Viimeistään käsitteellisen mallinnuksen jälkeen tietokannalle valitaan sopiva tietomalli. Käsitteellinen malli käännetään valitun tietomallin mukaisesti loogiseksi kuvaukseksi. [22]

Relaatiomallia käytettäessä käsittekaavion käsitteet käännetään lähes mekaanisesti relaatioiksi ominaisuuksineen ja niille valitaan pääavaimet. Pääavain valitaan joko käsitteille määritetyistä avaimista tai relaatiolle luodaan keinotekoinen pääavain eli surrogaatti. Surrogaatti on tietokannan automaattisesti generoima luku. Surrogaatteja käytetään yleisesti, koska niitä ei tarvitse päivittää, mikä on tietokannan eheyden ja tehokkuuden kannalta toivottavaa. [22]

Loogisessa suunnittelussa voidaan joutua suorittamaan myös normalisointi. Normalisoinnilla tarkastetaan tietokannan tietojen rakennetta. Sillä pyritään saamaan tiedot sellaiseen tilaan, että tietokannassa ei olisi toisteista tietoa.

### 2.2.4 Fyysinen suunnittelu

Fyysisessä suunnittelussa valitaan käytettävä tietokannan hallintajärjestelmä ja relaatiotietokannan tapauksessa tehdään tietokantaan taulut ja hakemistot. Hakemisto on tiettyjen hakujen kannalta otollisesti järjestetty ja sopiva tietorakenne. Se suunnitellaan aina tiettyihin hakuihin, jolloin sillä päästään tehokkaasti käsiksi näiden hakujen vaatimaan tietoon. Ilman hakemistoa tietokannasta haetaan tietoa aina

levyltä tauluja läpikäymällä, mikä on hidasta jatkuvasti tehtävien levyoperaatioiden vuoksi. [26]

Hakemisto määritellään yhdestä tai useammasta taulun sarakkeesta, ja taulun sarakkeita voi käyttää useammassa hakemistossa. Tavallisesti vähintään taulun pääavaimesta ja vierasavaimista tehdään hakemistot. Hakemistojen haittapuoli on niiden päivitystarve, kun tauluja päivitetään. Jos tietokannan taulu ei ole kovin suuri ja päivityksiä tehdään selvästi useammin kuin hakuja, hakemistolle ei ole välttämättä tarvetta. [22, 26]

### 2.2.5 Takaisinmallinnus

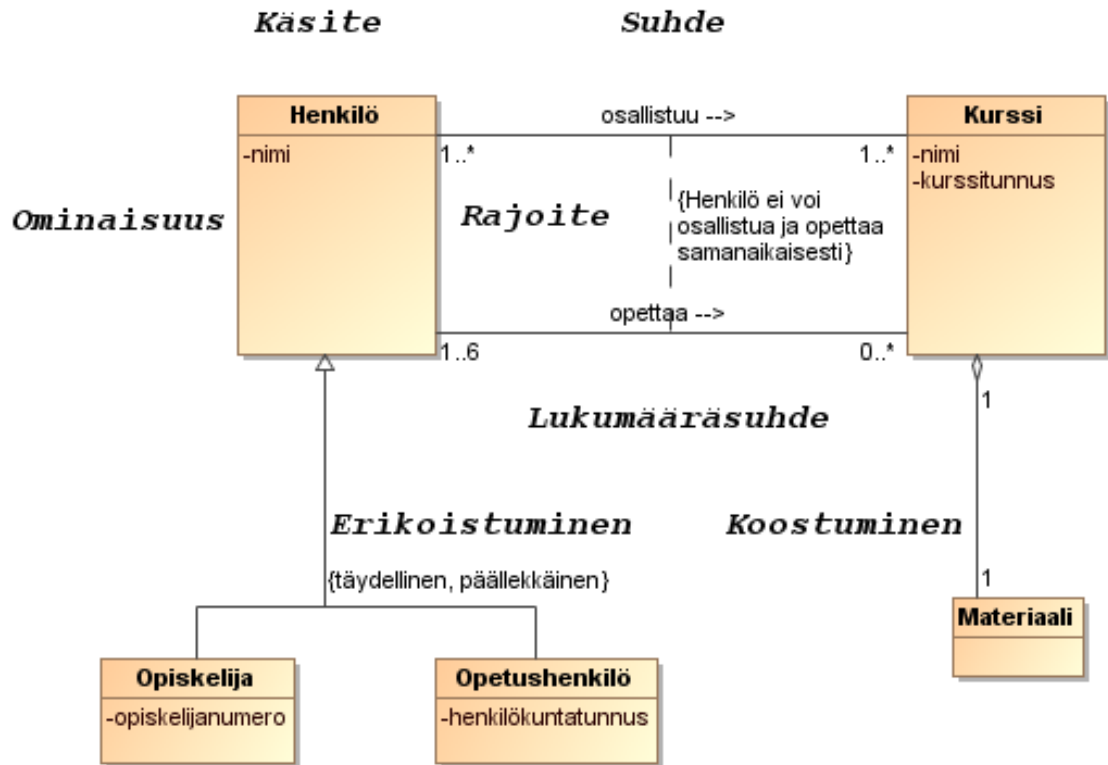
Takaisinmallinnuksessa suoritetaan tietokannan suunnittelussa toteutetut vaiheet päinvastoin. Tietokannan taulurakenne mallinnetaan takaisin relaatioiksi, jotka tarvittaessa mallinnetaan edelleen vielä käsitteelliseksi malliksi. Takaisinmallinnus tehdään yleensä silloin, kun huomataan tietokannan heikon suorituskyvyn johtuvan tietokannan huonosta rakenteesta. [22]

## 2.3 Käsitekaavio

Työssä käytetty kaaviotekniikka on Object Management Groupin standardoiman Unified Modeling Language (UML) [27] luokkakaavio. UML on standardoitu graafinen mallinnuskieli. Sen eri kaaviotyypit ovat yleisesti käytössä ohjelmistotuotannossa.

UML:n eri kaavioilla voidaan esittää muun muassa kuvattavan järjestelmän rakennetta, toimintaa ja tilamuutoksia. Luokkakaavion notaatiosta on esitetty esimerkki kuvassa 2.4. Tietokantasuunnittelussa luokkakaaviota käytetään yleisesti käsitteiden ja niiden välisten suhteiden esittämiseen. Tässä työssä käsitteet ja luokat kirjoitetaan selvyden vuoksi isolla alkukirjaimella.

Luokkakaaviossa käsite kuvataan luokkana, jolla on käsitteen ominaisuudet. Kuvassa 2.4 on kuvattu kohdealue, jossa on käsitteitä, ku-



Kuva 2.4: UML:n luokkakaavion notaatio

ten Henkilö ja Kurssi. Henkilöllä on ominaisuutena nimi. Kurssilla on ominaisuutena nimi ja kurssitunnus. Näillä käsitteillä on kaksi suhdetta. Henkilö voi osallistua Kurssille tai opettaa Kurssilla.

Lukumääräsuhteet määräävät, että Henkilö voi osallistua rajattomalle määrälle Kurseja, tai olla osallistumatta yhdellekään Kurssille (0...\*). Kurssille voi osallistua rajaton määrä Henkilöitä, mutta vähimmäismäärä on yksi (1...\*). Henkilö voi opettaa rajattomalla määrällä Kurseja tai ei yhdelläkään (1...\*). Kurssilla opettaa yhdestä kuuteen Henkilöä (1...6). Henkilön ja Kurssin suhteilla on rajoite, joka määrittää, että Henkilö ei voi osallistua Kurssille ja opettaa samaan aikaan samalla Kurssilla.

Henkilöstä erikoistuu kaksi käsitettä, Opiskelija ja Opetushenkilö. Näillä käsitteillä on yläkäsitteen Henkilö ominaisuudet ja lisäksi omat ominaisuutensa. Opiskelijalla on opiskelijanumero ja Opetushenkilöllä henkilökuntatunnus. Erikoistumisessa ei ole lukumääräsuhteita vaan

erikoistumisrajoitteet. Rajoitteita ovat täydellinen (complete), epätäydellinen (incomplete), päällekkäinen (overlapping) sekä epäjatkuva (disjoint). Täydellinen-rajoite määrittää, että Henkilö-käsitteen ilmentymät ovat aina siitä erikoistuneita käsitteitä eli joko Opiskelijoita tai Opetushenkilökuntaa. Täydellinen-rajoitteen vaihtoehto epätäydellinen-rajoite määrittäisi, että Henkilö-yläkäsitteestä voisi olla ilmentymiä ilman erikoistamista.

Päällekkäinen-rajoite määrittää, että Opiskelija voi olla samaan aikaan myös Opetushenkilö ja päinvastoin. Tämä tarkoittaa, että erikoistuneella käsitteellä voi olla yhden tai useamman toisen erikoistuneen käsitteen ominaisuudet. Päällekkäinen-rajoitteen vaihtoehto epäjatkuva-rajoite määrittäisi, että yläkäsitteestä voidaan erikoistaa käsite vain tasan yhdeksi käsitteeksi.

Kurssin suhde Materiaaliin ei ole tavallinen, vaan sitä kutsutaan Koostumiseksi. Koostumisessa koostuva käsite (Materiaali) on sidoksissa sen omistavaan käsitteeseen (Kurssi). Koostuva käsite on kuitenkin itsenäinen, eli sen olemassaolo ei ole sidoksissa omistavaan käsitteeseen.

## 2.4 Käytetyt tukisovellukset

### 2.4.1 MySQL

MySQL [28] on Oraclen omistama, yksi maailman käytetyimmistä avoimen lähdekoodin tietokannan hallintajärjestelmistä. MySQL on alun perin David Axmarkin, Allan Larssonin ja Michael ”Monty”Widenuksen 1990-luvulla kehittämä ja MySQL AB:n omistama ohjelmisto, joka yritysostojen [29, 30] myötä MySQL päätyi Oraclen omistukseen. MySQL on relaatiomallia käyttävä tietokannan hallintajärjestelmä ja käyttää SQL:ää tietokantakielenä. [31]

Taulukossa 2.1 on esitelty MySQL:n muutamien erityyppisten tietokantamoottoreiden ominaisuuksia. MyISAM on käytössä oletuksena. Sillä tiedon tallennus ja haku on nopeaa, mutta samanaikaiseen käyt-

töön ei ole tarjolla transaktioita. MyISAM ei tue myöskään vierasavaimia, joten eheyden ylläpito ei ole mahdollista tietokannan puolesta. InnoDB tukee vierasavaimia ja samanaikaiseen käyttöön lukituksia, joita voidaan käyttää tauluihin yksittäisten rivien tarkkuudella. MEMORY on tietokantamoottori, joka tallentaa taulut keskusmuistiin ja on siten sopiva väliaikaisen tiedon säilytykseen. NDBCLUSTER:ia käytetään hajautettuihin tietokantoihin, joissa taulut on hajautettu useisiin fyysisiin tietokoneisiin saavutettavuuden ja varmuuden parantamiseksi. [32]

Taulukko 2.1: MySQL:n joidenkin erityyppisten tietokantamoottorien ominaisuuksia. [32]

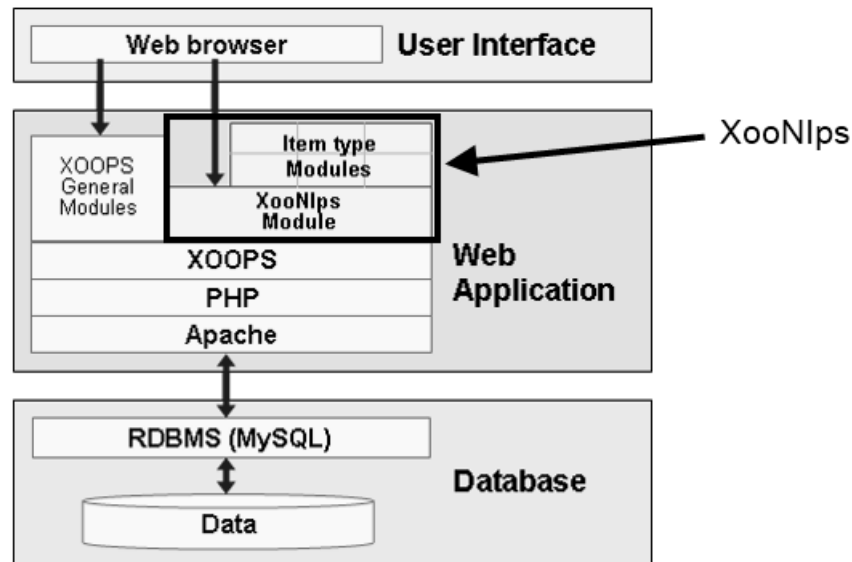
Tyyppi	Transaktiot	Vierasavaimet	Tallennusmedia	Hajautus
MyISAM	Ei	Ei	Levy	Ei
InnoDB	Kyllä	Kyllä	Levy	Ei
MEMORY	Ei	Ei	Keskusmuisti	Ei
NDB(Cluster)	Ei	Ei	Levy	Kyllä

## 2.4.2 XOOPS

XOOPS on avoimen lähdekoodin web-sovellusalusta ja sisällönhallintasovellus, jonka avulla voidaan toteuttaa esimerkiksi yhteisöllisiä web-sivustoja, intranetejä, web-portaaleja ja web-blogeja [33]. Sitä on kehitetty vuoden 2001 joulukuusta asti. XOOPS on toteutettu web-sovellusten tekoon tarkoitettulla PHP-ohjelmointikielellä ja se käyttää MySQL-relaatiotietokantaa tiedon tallennukseen. XOOPS on modulaarinen, eli sen päälle toteutettujen moduulien avulla omaa sivustoa voi muokata melko vapaasti. Siinä on kuitenkin valmiina perusmoduulit, joilla järjestelmää voidaan käyttää suoraan ilman omia moduuleja. Valmiiden tai itse tehtyjen moduulien asennus ja aktivointi on tehty yksinkertaiseksi. [34]

Kuvassa 2.5 on esitelty XOOPS:n korkean tason arkkitehtuuri, joka on kolmikerroksinen, tietokantaa käyttävän web-sovelluksen arkkitehtuuri [22]. Sen osat ovat käyttöliittymä (*User Interface*), sovelluslogiikka (*Web Application*) ja tietokanta (*Database*). XOOPS:n käyt-

töliittymä on web-selain, joka käyttää XOOPS:n sovelluslogiikkaa, eli valmiita moduuleja sekä niiden rinnalle tehtyjä omia moduuleja sisällön näyttämiseen.



Kuva 2.5: XOOPS:n korkean tason arkkitehtuuri [35]

Kuvassa näkyy XooNips esimerkkinä itse tehdystä moduulista. PHP:n (ohjelmointikieli) ja Apachen (palvelinsovellus) kautta ollaan yhteydessä RDBMS:ään (relational database management system, relaatiotietokannan hallintajärjestelmä), joka hoitaa tietokannan käsittelyn.

### 2.4.3 XooNips

XooNips on INCF:n Japanin yksikön (NIJC, Neuroinformatics Japan Center) kehittämä moduuli XOOPS:iin [36]. Se on kehitetty neurotutkijoiden tarpeita silmälläpitäen järjestelmäksi, jonne käyttäjät voivat tallentaa omia tiedostojaan ja halutessaan jakaa niitä järjestelmän muille käyttäjille. XooNips:a on kehitetty vuodesta 2005 alkaen ja se on edelleen kehityksessä. Kuvassa 2.5 näkyy XooNips:n liityntä XOOPS:iin ja millä tavalla se on osa kolmikerroksista web-sovellusten arkkitehtuuria.

XooNIps käyttää tiedon tallennukseen XOOPS:n tietokantaa. Moduulit, XooNIps mukaan lukien, laajentavat XOOPS:n tietokantaa omien tarpeidensa mukaisiksi. XooNIps luo tietokantaan omat taulut ja hakemistot tietosisältönsä käyttämään tietoa varten.

Japanin neuroinformatiikan yksikön tutkijat kehittivät sovelluksen alun perin omaan käyttöönsä, mutta kuitenkin sellaiseksi, että sitä on mahdollista laajentaa tarpeidensa mukaan. XooNIps:n tietokannasta tai sen suunnittelusta ei ollut tarjolla minkäänlaista dokumentaatiota. Apua tietokannan rakenteen selvittämiseksi saatiin Japanin neuroinformatiikan yksikön johtajalta sekä sovelluksen kehittäjäryhmän jäseniltä [37, 38].



## 3. TIETOKANNAN TARVE TUTKIMUSRYHMÄSSÄ

Tässä luvussa kerrotaan tietokannan tarpeesta TTY:n Signaalinkäsittelyn laitoksen M<sup>2</sup>oBSI-ryhmässä ja mitä vaatimuksia sekä tavallisia käyttötapauksia tutkimustiedon tietokannalle on.

### 3.1 Tietokannan tarve Signaalinkäsittelyn laitoksella

#### 3.1.1 M<sup>2</sup>oBSI-ryhmä

M<sup>2</sup>oBSI (Methods and Models for Biological Signals and Images) on signaalinkäsittelyn laitoksella toimiva tutkimusryhmä. Se keskittyy biologisten ja lääketieteellisten kuvien rekonstruointiin, käsittelyyn ja analysointiin. Ryhmän tavoitteena on kehittää kolmiulotteisten toiminnallisten kuvien analysointiin tarkoitettuja automaattisia menetelmiä. [39]

Tutkimusryhmän ongelma on tiedon hallinnan organisointi. Tutkijat säilövät tietoja yhteiselle verkkolevyille, mutta eivät mitenkään organisoidusti, vaan omiin hakemistoihin parhaaksi katsomallaan tavalla. Tällainen toimintatapa hyödyttää yksittäistä tutkijaa henkilökohtaisesti, mutta ryhmän näkökulmasta se ei ole järkevä. Tutkijoiden voi olla hankala löytää tarvitsemaansa tietoa toistensa hakemistoista, koska hakemistojen tieto ei ole järjestyksessä yhteisesti sovitulla tavalla. Ongelma tulee ilmi erityisesti silloin, kun tutkijoita lähtee pois ryhmästä ja tämän jälkeen ryhmässä on tarvetta käyttää poislähteneiden tutkijoiden tutkimustietoa.

Tämän ongelman vuoksi ryhmän johtaja näki tarpeen ottaa käyttöön sovellus, jonka avulla ryhmän jäsenet voisivat tallentaa tutki-

muksessa syntyvää tietoa organisoidusti. Tietokanta sopii hyvin tiedon organisoituun tallennukseen. Tietokanta ei yksin kuitenkaan riitä, vaan sen on oltava osa sovellusta, jolla tietoa tallennetaan. Sovelluksen olisi tarjottava vähintään käyttöliittymä, jolla tietotekniikasta vähemmän ymmärtävät tutkijatkin pystyvät helposti käyttämään tietokantaa työssään. Tutkimusryhmän omilla resursseilla tällaisen sovelluksen tekemistä alusta asti ei harkittu.

Kokeiltavaksi sopiva sovellus löytyi muualta. Tutkimusryhmän vetäjä on INCF:n Suomen yksikön vastuuhenkilö, ja tämän roolin myötä hän on ollut yhteyksissä Japanin yksikön vastaavaan johtajaan. Japanin yksikössä kehitettiin XooNIps-niminen sovellus (4.1) tutkijoiden tutkimustiedon säilömiseen ja tämä kuulosti sopivalta sovellukselta kokeiltavaksi tutkimusryhmässä.

### 3.1.2 Tutkimustieto tutkimusryhmässä

Tutkimusryhmän tutkimuksessa käytettävä tieto on vaihtelevaa erilaisten tutkimuskohteiden myötä. Tästä johtuen käsiteltävä tieto on vaihtelevaa eikä siitä voi tehdä monia oletuksia. Tiedossa yhdenmuukaista on sen numeerisuus. Lähes kaikkea tutkimustietoa voidaan käsitellä numeerisessa muodossa esimerkiksi Matlab-ohjelmalla. Kuvatietoa on paljon esimerkiksi PET-tyyppisenä.

Käytetyimpiä tiedostotyyppejä PET:ssä ovat ECAT [40], DICOM [41] sekä Analyze [42]. ECAT on positronien kuvannukseen suunniteltu tiedostotyyppi, jossa on kuvakoko sekä kameran identifioiva tyyppi. DICOM on digitaalisen lääketieteellisen kuvannuksen ja tiedonvälityksen standardi, joka käsittää muun muassa DICOM-laitteiden välisen tiedonvälityksen ja oman tiedostotyyppin. Tiedostotyyppi sisältää kuvaosan, joka tukee suurimpaa osaa lääketieteellisistä kuvatiedostotyypeistä sekä otsikon metatiedoille. Analyze on kuvatiedostotyyppi, joka käsittää kaksi tiedostoa, itse kuvatiedoston sekä otsikkotiedoston metatiedoille.

Koska ryhmän tavoite on kehittää biologisten ja lääketieteellisten kuvien käsittelyä, kuvat ovat myös keskeisin osa ryhmän tutkimustietoa. Kuvia on sekä 2D- että 3D-muodossa, ja molempia myös aikasarjoina. Ryhmän käyttämän tiedon määrä on suuri, teratavujen luokkaa, koska käsittelyssä on nimenomaan useampiulotteisia kuvia ja näitä sarjallisina aikatasossa.

Ryhmässä käsitellään toki myös muuta tietoa. Lähdekoodeja, simulaatioita, perinteisiä dokumentteja ja erilaista mittaustietoa eri muodoissa (esimerkiksi aikasarjatietoa) käytetään tutkimuksessa, ja näiden tallennukselle on yhtä lailla tarvetta. Näissä kaikissa tiedostotyypit vaihtelevat, eikä mitään tiedostotyyppiä ole valittu erityisesti priorisoitavaksi. Tosin aiemmin mainitut tiettyjen laitteiden tukemat omat tiedostotyypit eivät ole käytössä, koska näille laitteille tarkoitettuja ohjelmistoja ei ole käytössä. Tietokantanäkökulmasta voidaan olettaa tiedon olevan käytännössä minkälaista tahansa.

### **3.1.3 Tietokannan vaatimukset**

Kuten luvuissa 1.1.3 ja 3.1.2 esitetään, tietokannassa halutaan säilyttää hyvin heterogeenistä tietoa. Tallennettavien tiedostojen koko ja formaatti ovat vaihtelevia. Pääasiassa tallennettava tieto on tekstistä koostuvia eri formaateilla tallennettuja tiedostoja, mutta myös kuva- ja videotiedostoja halutaan tallentaa. Tallennettavasta tiedosta halutaan tallentaa myös metatietoja. Joissain tiedostoformaateissa metatiedot ovat erillisessä otsikkotiedostossa, jolloin tämä tiedosto on pystyttävä tallentamaan tietokantaan varsinaisen tiedoston tallennuksen yhteydessä. Tietoa on voitava lisäksi linkittää toisiinsa. Kun esimerkiksi haetaan jonkin tiedoston otsikkotiedosto (header), sen on viitattava kuvailemaansa tiedostoon ja toisinpäin.

Henkilöt, jotka käyttävät järjestelmää, voivat tallentaa tietokantaan tiedostoja. Nämä tiedostot voivat olla joko ainoastaa henkilölle itselleen näkyviä tai ne voivat näkyä myös muille. Henkilöt voidaan jakaa

eri ryhmiin, jotta järjestelmään saadaan esimerkiksi tutkimusryhmiä vastaavat joukot. Järjestelmässä ryhmälle voidaan luoda hakemistoja, joissa olevat tiedostot ovat vain ryhmään kuuluvien henkilöiden haettavissa. Ryhmälle asetetaan vähintään yksi moderaattori, joka hallinnoi ryhmän käyttäjälistaa ja hakemistoja. Koko järjestelmälle asetetaan vähintään yksi ylläpitäjä, joka hallinnoi koko järjestelmän tietoa sekä kaikkia henkilöitä ja ryhmiä järjestelmässä. Tällä tavoin hierarkia saadaan pidettyä matalana.

### 3.1.4 Käyttötapaukset

Tietokannalle on useita käyttötapauksia. Niitä kaikkia ei käydä läpi, mutta tavallisimmat käyttötapaukset selitetään alla. Niitä käytetään luvussa 5 tietokannan suunnittelussa saantipolkujen analysoinnissa.

**Käyttötapaus 1:ssä** tutkija tallentaa valmiin julkaisun. Hän haluaa tallentaa julkaisun paikkaan, josta hänen tutkimusryhmänsä jäsenet voivat hakea julkaisun omaan käyttöönsä. Myös julkaisussa käytetty tieto (esimerkiksi kuvat, mallit ja simulaatiot) halutaan saataville. Tällä tavalla tutkijan työ ei jää vain hänen omiin arkistoihin vaan hän voi jakaa työnsä myös kollegoidensa käyttöön.

**Käyttötapaus 2:ssa** tutkija tarvitsee kollegansa tietyssä julkaisussa käytettyjen kuvien mittaustiedon omaan julkaisuunsa. Hän etsii ensin kollegansa julkaisun ja tätä kautta kuvien mittaustiedon. Julkaisu ja siinä käytetty tieto on vapaasti tutkijaryhmän käytettävissä, joten tutkija voi käyttää tietoa omassa julkaisussaan.

**Käyttötapaus 3:ssä** Tutkija tarvitsee julkaisut, joissa on käytetty tiettyä simulaatiotulosta. Hän etsii simulaatiotuloksen nimellä kaikki julkaisut, joissa tulosta on käytetty.

**Käyttötapaus 4:ssä** Tutkija hakee avainsanalla julkaisuja, joissa sana esiintyy.

## 4. XoonIPS

Tässä luvussa kerrotaan Tampereen teknillisen yliopiston Signaalinkäsittelyn laitoksella käyttöönotetusta XoonIps-sovelluksesta, lyhyesti sen käyttöönotosta sekä muokkauksesta. Luvussa esitellään ja analysoidaan myös XoonIps:n tietokantaa.

### 4.1 XoonIps-sovelluksen esittely

XoonIps on tiedon organisointiin tarkoitettu, tietokantaa käyttävä sovellus. Se on XOOps-sisällönhallintasovelluksen (2.4.2) moduuli, joka tarjoaa käyttöliittymän tiedostojen tallennukseen ja hakuun. XoonIps käyttää sille kehitettyjä moduuleja erilaisten tiedostotyyppien esitykseen. Tiedostotyyppin ilmentymä on sovelluksessa nimeltään Item. Itemtyypit on valittu lääketieteellisen tutkimuksen tutkijoille tärkeiden tiedon tyyppien mukaan.

XoonIps:n item-tyypit on koottu taulukkoon 4.1. Erikseen mainittava tyyppi on Paketointi-Item (Binder), joka ei esitä mitään tallennettavan tiedon tyyppiä, vaan on tarkoitettu paketoimaan useampia item-viittauksia yhden itemin sisälle.

XoonIps:ssa item tallennetaan tiettyyn hakemistoon, joka on XoonIps:ssa nimeltään Indexi (Index). Indeksit ovat puurakenne, eli niissä voi olla useita sisäkkäisiä Indeksejä. Juuritasolla on kolme erityyppistä indeksiä. Yksityinen indeksi (Private Index) on käyttäjän oma, ja sinne laitettut itemit näkyvät vain käyttäjälle itselleen (ellei itemeitä ole laitettu myös muualle). Julkisen indeksin (Public Index) sisältö näkyy kaikille käyttäjille. Jos käyttäjä kuuluu johonkin ryhmään, tälle ryhmälle voidaan luoda indeksi (Group Index), jonka sisältö näkyy vain ryhmän jäsenille.

Taulukko 4.1: XooNIps:n Item-tyypit

Item-tyyppi	Selite
Binder	Itemien paketoimiseen tarkoitettu tyyppi
Book	Kirja
Conference	Konferenssipaperi
Data	Numeerinen tieto teksti/numero/kuvamuodossa
File	Vapaasti valittava tiedostomuoto
Memo	Oma muistio
Model	Malliohjelma
Paper	Julkaisu
Presentation	PowerPoint-esitys (tai vastaava)
Simulator	Simulaatio-ohjelma
Stimulus	Kokeellinen tieto
Tool	Työkaluohjelma
URL	Internet-linkki

Itemille annetaan tallennettaessa tunniste ja nimi sekä joitain muita pakollisia tietoja riippuen sen tyylistä. Edelleen tyylistä riippuen itemille voidaan antaa lisäksi tarkentavia tietoja. Itemille voidaan asettaa myös viittauksia toisiin itemeihin.

## 4.2 XooNIps:n käyttöönotto ja muokkaus

XooNIps asennettiin TTY:n Signaalinkäsittelyn laitoksen ylläpidon tarjoamalle web-palvelimelle. Pääsy ohjelmaan rajoitettiin vain tutkimusryhmän jäsenien koneille. XooNIps vaatii toimiakseen sovelluspalvelimen sekä MySQL-tietokannan. TTY:n Signaalinkäsittelyn laitoksen ylläpidon tarjoamalla palvelimella oli käytettävissä MySQL-tietokanta sekä yleisesti käytetty Apache Tomcat -web-säiliö. XooNIps asennettiin tähän palvelimelle, mutta sovellus ei ollut vielä valmis käyttöön.

XooNIps vaati muokkausta, jotta siitä sai ryhmän vaatimukset (3.1.3) täyttävän sisällönhallintasovelluksen. XooNIps:ssa ei ole mahdollista tehdä kahdensuuntaisia viittauksia itemeiden välille automaattisesti. Jos esimerkiksi julkaisun ja siinä käytetyn kuvan välille halutaan viittaus molempiin suuntiin (julkaisussa viittaus kuvaan ja kuvassa viit-

taus julkaisuun), ne on tehtävä käsin. Tämä on aikaa vievää ja turhauttavaa käyttäjälle, varsinkin jos viitattavia itemeitä on paljon.

Niinpä XooNIps:n lähdekoodia muokattiin XooNIps:n kehittäjäryhmän jäsenen avulla siten, että kahdensuuntainen viittaus luodaan automaattisesti kahden itemin välille, kun niiden välille luodaan viittaus ensimmäisen kerran. Muutos oli koodimäärään nähden pieni, vain joi-tain kymmeniä rivejä.

### 4.3 XooNIps:n tietokanta

XOOPS:n tietokannassa taulut tallennetaan MyISAM-tietokantamoottorilla, jossa ei ole vierasavaimia käytettävissä. Tästä syystä XooNIps:n kehittäjät päätyivät käyttämään MyISAM:a osaan XooNIps:n tauluista. Lopuissa on käytössä InnoDB-tietokantamoottori. Molemmilla tietokantamoottoreilla tallennettuja tauluja käytetään sekaisin ja tämä voi aiheuttaa ongelmia tietokannanhallintajärjestelmän toiminnassa. Sen takia vierasavaimia ei ole käytetty missään XooNIps:n tauluista.

Täten tietokannan eheyttä ei ole mahdollista ylläpitää tietokannanhallintajärjestelmän avulla automaattisesti, vaan jokainen taulu on eristyksissä muista tauluista, vaikka useammassa taulussa on samaa tietoa. Päivitykset tietokantaan on siis tehtävä ohjelmakoodissa moneen kertaan, jos samaa tietoa on useammassa taulussa. XooNIps:n tauluissa on määritelty indeksit sarakkeille, joita käytetään SELECT-lauseissa useasti.

XooNIps:n tietokannassa on 48 taulua ja lisäksi 23 taulua eri itemtyypeille. Tässä on käsitelty vain pientä osaa tauluista. Käsitellyn osan on katsottu olevan työn analyysin kannalta tärkein, eli miten itemien ja indeksien tiedot jakautuvat eri tauluihin, ja miten itemien ja indeksien yhteys toisiinsa on tehty.

#### 4.4 XooNIps:n tietokannan takaisinmallinnus käsitteelliseksi malliksi

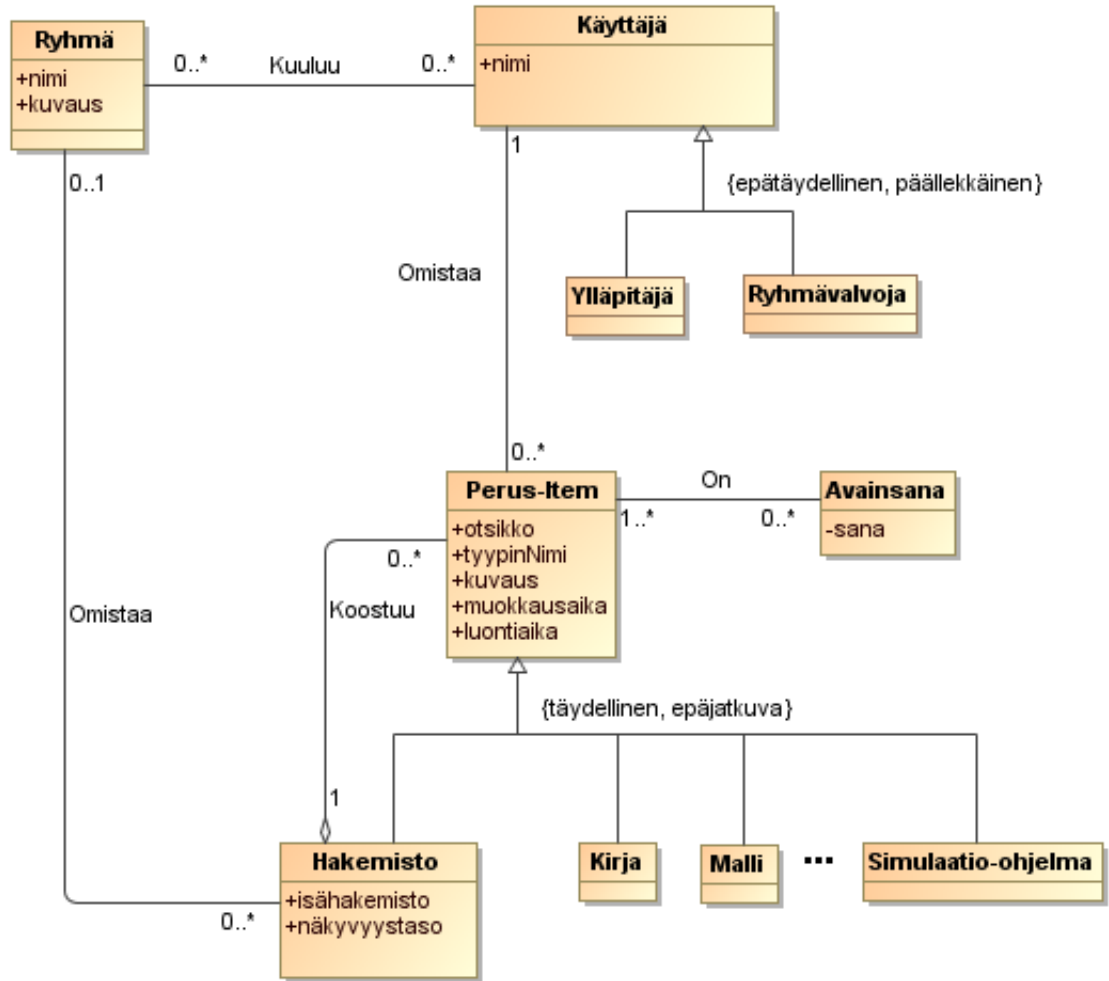
XooNIps:n tietokannalle tehtiin takaisinmallinnus, jotta sitä voitiin analysoida ja vertailla suunniteltuun ideaaliseen malliin. Vierasavainten puuttumisen takia tietokannan taulurakennetta ei pystynyt selvittämään taulujen luontilauseiden avulla. Taulujen suhteet selvitettiin käymällä läpi XooNIps:n koodia ja tarkistamalla sieltä SQL-lauseiden sisältö, josta selvisi minkä taulujen hakemistoja tai sarakkeita verrattiin toisiinsa WHERE-lauseissa. Lisäksi XooNIps:n kehittäjäryhmän jäsenen antamat tiedot taulurakenteesta tukivat selvitystyötä.

Rakenteen käsittelyssä tietokannasta rajattiin kaikki kiinnostavat taulut takaisinmallinnusta varten. Tähän otettiin mukaan tauluja, jotka liittyvät käyttäjiin, käyttäjäryhmiin sekä XooNIps:n itemeihin ja indekseihin. Taulujen yhteydet toisiinsa selvitettiin ja tällä tavalla mallinnettiin, millaisia käsitteitä tietokannan tässä rajatussa osassa on. Kuvassa 4.1 on esitetty XooNIps:n tietokannan käsittekaavio takaisinmallinnuksen jälkeen.

Käsittekaaviossa Perus-Item kuvaa XooNIps:n peruskäsitettä. Siitä erikoistuu kaikki sovelluksen käsittämät tietotyypit (itemit), kuten kuvassa 4.1 esimerkkinä olevat Kirja, Malli ja Simulaatio-ohjelma. Kuvassa ei ole kaikkia käsitteitä XooNIps:n tietotyypeistä, ne on koottu erikseen kuvaan 4.2 (ja niitä vastaavat itemit on esitelty taulukossa 4.1).

Perus-Itemistä erikoistuu myös Hakemisto, joka vastaa XooNIps:n indeksiä. Tämä ratkaisu käyttää rekursiokooste-suunnittelumallia [43]. Hakemisto on Perus-Item, joka voi omistaa muita Perus-Itemejä. Näin rakenteesta tulee joustava, koska Hakemisto koostuu sellaisista Perus-Itemeistä, joiden tyyppiä ei tarvitse määrittää rakenteessa. Hakemiston ominaisuus isähakemisto määrittää, mikä Hakemisto on sen isä hakemistohierarkiassa. Näkyvyystaso taas määrittää keille kaikille Hake-

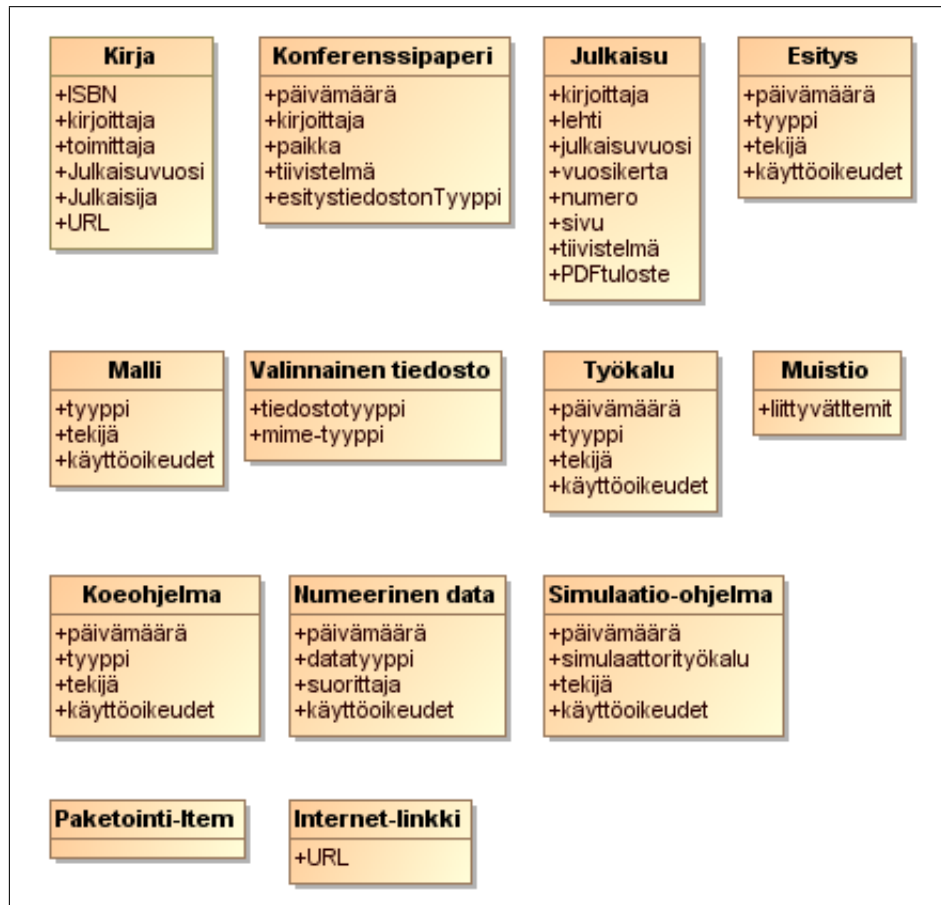




Kuva 4.1: XooNIps:n tietokannan takaisinmallinnettu käsitteellinen malli

misto näky. Se voi näkyä kaikille käyttäjille, vain ryhmään kuuluville käyttäjille tai ainoastaan Hakemiston omistavalle käyttäjälle.

Perus-Itemillä on yksi omistaja, Käyttäjä, joka voi muokata Perus-Itemiä tai poistaa sen. Perus-Item sijaitsee Hakemistossa, jonka näkyvyys muille Käyttäjille määräytyy tämän Hakemiston tyypin mukaan (julkinen, ryhmä, yksityinen). Poikkeustapaus Perus-Itemeissä on Hakemisto, jonka omistaja voi olla Käyttäjän sijaan myös Ryhmä. Ryhmään kuuluvat käyttäjät pääsevät katsomaan ryhmän omistavia Hakemistoja ja niiden sisältöä. Jos Käyttäjä on Ryhmänvalvoja tiettyssä Ryhmässä, hän voi muokata sen Hakemistoja. Ylläpitäjä asettaa Ryhmille Käyttäjät, hyväksyy uudet Käyttäjät sekä ylläpitää julkista Hakemistoa.



Kuva 4.2: XooNIps:n käsitteellisen mallin tietotyypit ja niitä kuvaavat käsitteet

Kuvassa 4.2 on esitetty XooNIps:n item-tyyppejä vastaavat käsitteet. Näissä voidaan havaita tiettyjä yhtäläisyyksiä. Esimerkiksi Kirjalla, Konferenssipaperilla ja Artikkelilla on kaikilla tiedot julkaisua-jankohdasta. Mallilla, Simulaatio-ohjelmalla, Esityksellä, Työkalulla, Koeohjelmalla ja Numeerisella datalla on samoina ominaisuuksina tekijä sekä käyttöoikeudet. Kirjassa ja Artikkelissa tekijä on korvattu kirjoittajalla.

Käsitteissä on myös tiettyjä yksilöllisiä ominaisuuksia, kuten Numeerisen datan tietotyyppi ja Simulaatio-ohjelman simulaattorityökalu.

## 5. XoonIps:n Evaluointi














Tässä luvussa evaluoidaan XoonIps-sovellusta. Evaluoinnin perustana käytetään XoonIps:n käyttökokeilujen tuloksia sekä luvussa 3 esiteltyjä vaatimuksia ja käyttötapauksia. Evaluoinnissa arvioidaan sovelluksen ominaisuuksien hyödyllisyyttä ja puutteita yleisesti sekä erikseen tietokannan osalta. XoonIps:n tietokannasta suunnitellaan uudempi malli, jota vasten XoonIps:n tietokantaa evaluoidaan. Suunnittelu sisältää käsitteellisen mallinnuksen sekä loogisen suunnittelun.

### 5.1 Käyttökokeilut XoonIps:lla

XoonIps:n käytöstä tehtiin kokeiluja muutamien testikäyttäjien kanssa TTY:n Signaalinkäsittelyn laitoksella. Heidän kanssa tarkasteltiin sovelluksen käyttöliittymän ulkoasua ja käytettävyyttä, sekä sovelluksen toimintaa. Tässä yhteydessä käytettiin sovelluksen vapaasti saatavilla olevaa versiota, muokattu versio ei ollut käytettävissä vielä. XoonIps:iin tallennettiin viisi eri artikkelia, mallia

Kuvassa 5.1 näkyvän XoonIps:n etusivun keskiosan asettelussa oli testikäyttäjien mukaani puutteita, sillä he eivät kokeneet sitä tarpeeksi intuitiiviseksi. Etusivun keskelle sijoitetut item-tyyppien linkit hämärsivät testikäyttäjiä, koska niistä ei pääse lisäämään itemiä, vaan ainoastaan kunkin item-tyypin listaussivulle. Lisäksi listaussivulla on näkyvillä oletuksena kaikki kyseisen tyyppin itemit.

Etusivun navigointivalikoiden jako kahteen reunaan oli koehenkilöiden mielestä sekava ratkaisu (kuva 5.2). Lisäksi he pitivät oikeanpuoleisen navigointivalikon linkkien asettelua epäloogisena.

Registered Itemtypes	
 <b>Binder</b> Binder	 <b>Book</b> Related Book Collection.
 <b>Conference</b> Electrical presentation files for conference. PowerPoint / PDF / Illustrator / Other	 <b>File</b> Various types of File. ini / dat
 <b>Data</b> Result data in numerical text/image/movie formats. Excel / Movie / Text / Picture / Other	 <b>Memo</b> Personal Memo Pad.
 <b>Model</b> Model programs/scripts. Matlab / Neuron / OriginalProgram / Satellite / Genesis / A-Cell / Other	 <b>Paper</b> Related paper collection.
 <b>Presentation</b> Electrical presentation files. PowerPoint / Lotus / JustSystem / HTML / PDF / Other	 <b>Simulator</b> Programs/scripts for simulation. Matlab / Mathematica / Program / Other
 <b>Stimulus</b> Picture, movie and program files for experimental stimuli. Picture / Movie / Program / Other	 <b>Tool</b> Programs/scripts for data analysis. Matlab / Mathematica / Program / Other
 <b>Url</b> Link information.	

Kuva 5.1: XooNIps:n etusivun keskiosa

Itemien kahdensuuntaisten viittausten automaattisen toiminnan puute koettiin huonoksi asiaksi. Jos kahden itemin välille halutaan viittaukset molempiin suuntiin, on se tehtävä käsin. Item A:sta viittaus item B:hen saadaan suoraan, kun item A lisätään tietokantaan. Item B:n viittaus item A:han on tehtävä jälkikäteen muokkaamalla item B:n sivua. Tämä aiheuttaa liikaa vaivaa varsinkin tilanteessa, jossa item A:ta lisättäessä halutaan laittaa viittauksia esimerkiksi kymmeneen muuhun itemiin.

Paketointi-itemia ja itemien viittausominaisuutta testikäyttäjät pitivät päällekkäisinä. Yksi pahimmista puutteista oli heidän mielestä se, että item käsittää vain yhden tiedoston. Esimerkiksi simulaatiotuloksia tallennettaessa tämä ongelma tulee vastaan, sillä simulaatio-



Kuva 5.2: XooNIps:n etusivun vasemmanpuoleinen (a) ja oikeanpuoleinen (b) navigointivalikko

tulos voi olla jaettuna ajallisesti useaan tiedostoon, jolloin niitä kertyy kymmeniä tai jopa satoja. Jokaisen tiedoston lisääminen järjestelmään yksitellen olisi työmäärältään kohtuuttoman suuri. Kymmenien samaan simulaatioajoon pohjautuvien itemien listaus ja hakeminen olisi myös liian hankalaa.

Model-tyypiksi voidaan määrittää vain kehittäjien ennalta määrittelemiä mallityyppejä, tai liian suppeasti kuvaileva Muu-tyyppi. Testikäyttäjät kokivat tämä huonoksi ratkaisuksi ja parannusehdotus oli vapaasti määritettävä Modelin tyyppi Muu-tyypin sijaan. Lisäksi kai-

kille itemeille toivottiin kuvaa tiedoston sisällöstä suoraa itemin sivulle (esimerkiksi thumbnail-kuva), jotta tiedostoa ei tarvitsisi ladata omalle koneelle vain nähdäkseen millainen se on.

## 5.2 XooNIps:n arviointi

XooNIps täyttää pääasiassa sille asetetut vaatimukset. Sovelluksella pystyy tallentamaan haluamaansa tietoa ja jakamaan sitä muille sovelluksen käyttäjille. Muiden käyttäjien tallentamaa tietoa on mahdollista hakea sovelluksen kanssa, esimerkiksi julkaisuja. Julkaisuihin linkitetyt kuvat tai muut tiedostot ovat myös saatavilla julkaisun sivun yhteydessä. Haku onnistuu myös toisinpäin (haetaan julkaisut, joissa tiettyä kuvaa on käytetty), kunhan kahdensuuntainen viittaus on luotu tietojen välille.

Kahdensuuntaisen viittauksen luonti on XooNIps:ssa liian hankalaa, koska viittaus kahden itemin välille on tehtävä käsin molempiin itemeihin. Viittauksen luonti itemistä toiseen vie noin puoli minuuttia aikaa. Viiden itemin kesken viittaukset kaikista kaikkiin itemeihin veisi näin ollen 10 minuuttia (20 viittausta). Luvussa 4.2 esitelty XooNIps:iin tehty muokkaus puolittaa kahdensuuntaisen viittauksen luontiin kuluvan ajan. Kahdensuuntaisten viittausten luonti viiden itemin kesken veisi näin ollen 5 minuuttia.

Tiedon jaottelu on XooNIps:ssa liian rajoitettua. Tietotyypit on valmiiksi määritetyt ja muunlaista tietoa on järkevää tallentaa ainoastaan yleisen tiedostotyypin (File) alle. Tämä tietotyyppi on taas niin yleiseksi määritetty, että se ei sisällä juurikaan rakennetta metatietojen tallennukseen.

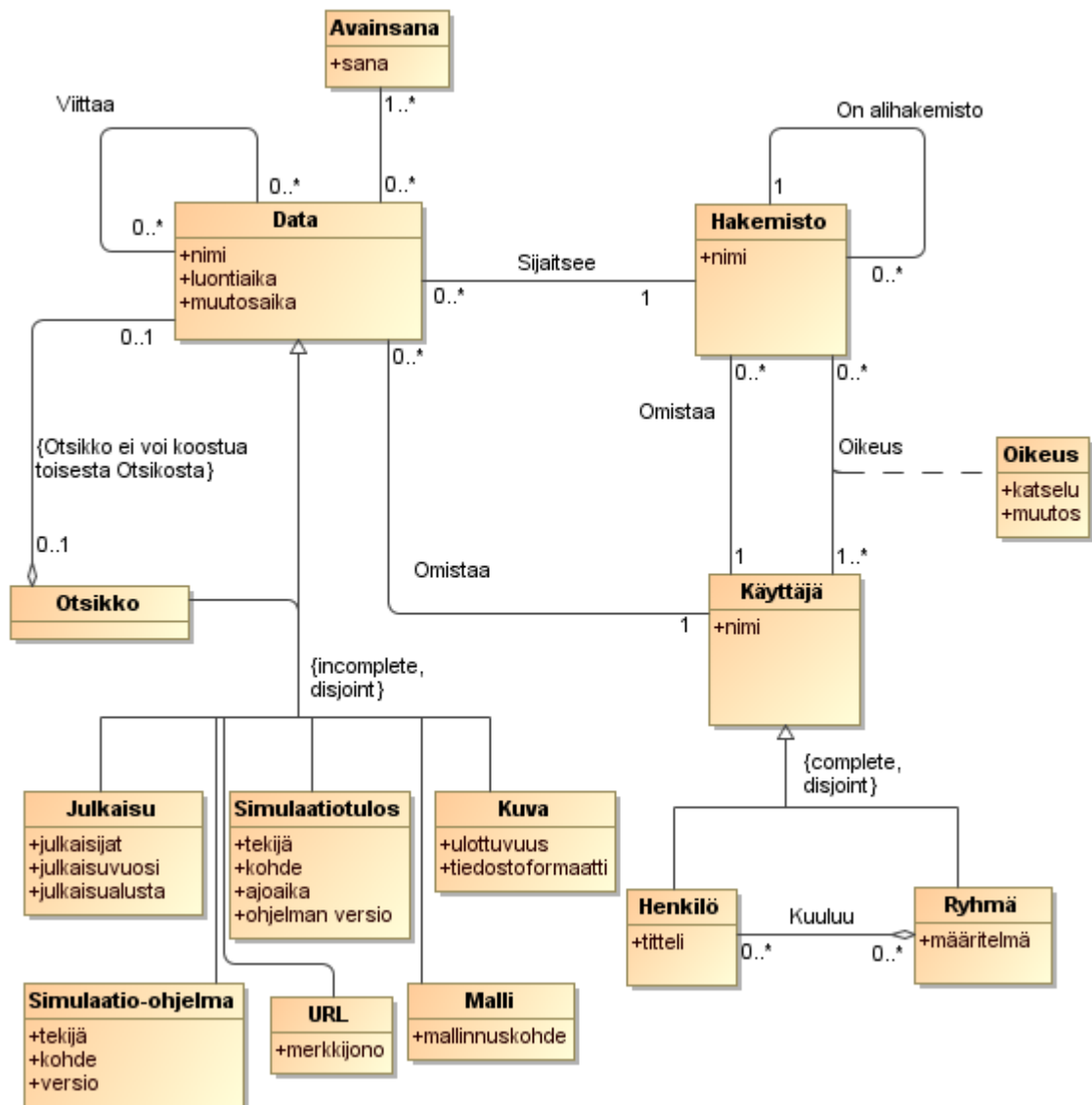
## 5.3 Tietokannan uudelleen suunniteltu malli

XooNIps:n tietokannan uudelleen suunnitellussa mallissa tehtiin ensin käsitteellinen mallinnus, joka aloitettiin suunnittelemalla käsittekaavio järjestelmän käsitteistä ja niiden suhteista. Saantipolkuanalyysi-

silla varmistettiin käsittekaavion oikeellisuus. Loogisella suunnittelulla käännettiin käsitteellinen malli relaatiomalliksi.

### 5.3.1 Käsittekaavio

Käsittekaaviosta pyrittiin tekemään mahdollisimman valmis, mutta suunnittelussa otettiin huomioon mahdollinen päivityksen tarve. Järjestelmään on pystyttävä tekemään myös muutoksia jälkeenpäin mahdollisimman pienellä vaivalla ja rakenteiden muutoksella. Käsittekaavio on esitetty kuvassa 5.3.



Kuva 5.3: Uudelleen suunnitellun mallin käsittekaavio.

*Data* kuvaa tietoa mitä järjestelmään tallennetaan. *Data* on yläluokka, yleiskäsite. Erikoistumista käyttämällä tietotyyppien jaottelu on selkeä, jolloin myös uusia tietotyyppettä on mahdollista lisätä järjestelmään jälkikäteen suoraan yläluokasta erikoistamalla. *Julkaisu*, *Simulaatiotulos*, *Simulaatio-ohjelma*, *Muistiinpano*, *Kuva* ja *Malli* ovat *Data*sta erikoistettuja alaluokkia. *Datan* tyyppi ei voi olla kuin yhtä alaluokkaa, mutta se ei välttämättä ole mikään alaluokista, eli järjestelmään tallennettava tieto voi olla myös yläluokan *Data* tyyppistä. Tähän ratkaisuun päädyttiin, koska järjestelmään voidaan haluta lisätä tyyppiltään sellaista tietoa, mitä sinne ei ole määritelty (edellä luetellut tietotyypit). Toisaalta tallennettava tieto ei aina edes vaa di sen tarkempaa erottelua tietyksi tietotyyppiksi, jolloin se voidaan määritellä järjestelmään vain *Data*ksi.

*Datalla* on ominaisuuksina nimi, luontiaika ja muutos aika. *Datan* aikaleimoilla pidetään kirjaa sen historiasta järjestelmässä, koska siihen voidaan tehdä muutoksia lisäyksen jälkeen. Lisäksi tiedolle on määritettävä myös vähintään yksi avainsana. Avainsanoilla *data* jaotellaan eri aihepiireihin, jolloin sen hakeminen ja jaottelu helpottuu. Avain sanat on hyvä määritellä etukäteen koko järjestelmälle ja pitää niiden määrä järkevänä, jotta käyttäjät voivat olla tietoisia mihin eri aihepiireihin tieto on jaettu.

Eri tietotyypeille on myös määritetty ominaisuuksia. *Julkaisuun* merkitään julkaisun tekijät, julkaisuvuosi sekä julkaisualusta (lehti, koelma, konferenssipaperi). *Simulaatio-ohjelmaan* ja *-tulokseen* merkitään simuloinnin kohde ja tekijä. Lisäksi *simulaatio-ohjelmaan* merkitään sen versio, jotta lähdekoodin muutokset saadaan versioitua selkeästi. *Simulaatiotuloksessa* on myös simulointia ajoaika sekä käytetyn *simulaatio-ohjelman* versio, jolla simulointi ajettiin. Tällä tavalla ollaan koko ajan tietoisia ohjelman versiosta, jolla tulokset on saatu eikä väärinkäsityksiä tapahdu. *Kuvaan* määritetään sen ulottuvuus sekä tiedostoformaatti, *Muistiinpanoon* asetetaan merkkijono ja *Malliin* mallinnettava kohdeilmiö tai -tapahtuma.



*Otsikko* on tietotyyppi, joka vastaa luvussa 3.1.3 esitetyissä vaatimuksissa mainittua otsikkotiedostoa, jolla on tietyn tiedoston metatietoja. Otsikko poikkeaa tavallisista tietotyypeistä, sillä se on koostumissuhteessa yläluokkaansa eli Dataan. Tavallinen tietotyyppi siis sidotaan Otsikkoon. On mahdollista myös, että Datalla ei ole Otsikkoa ollenkaan (eli otsikkotiedostoa tälle tiedostolle ei ole olemassa). Huomioitavaa on, että Datalla ei voi olla kuin yksi Otsikko eikä Otsikolla kuin yksi Data, jonka metatietoja se kuvaa. Merkityllä rajoitteella taas poistetaan väärinkäsityksen riski, eli otsikkotiedosto ei voi olla koostumissuhteessa toiseen otsikkotiedostoon. Kooste-suunnittelumallia käytetään, jotta Otsikko-tyyppinen tieto saadaan hierarkkisesti tavallisen tiedon yläpuolelle ilman, että tavallisen tiedon tietotyyppiä olisi määritettävä. Näin Otsikko pysyy rinnasteisena tietotyyppinä järjestelmän muihin määriteltyihin tietotyyppeihin, mutta hierarkia omistettavaan tietoon on silti olemassa. [43]

Data voi viitata suoraan toiseen Dataan (myös useampaan), millä saadaan toteutumaan tiedon linkitys keskenään. Näin saadaan rakennettua helposti linkkejä ja linkkiketjuja tietojen välille, jolloin tiedon hakeminen helpottuu. Data sijaitsee oletuksena tasan yhdessä Hakemistossa, jotta tietoa ei olisi monessa paikassa ja siten tietokannan eheyden ylläpito helpottuu. Hakemistoilla on keskenään hierarkkinen suhde eli Hakemistoilla voi olla toisia Hakemistoja alihakemistoina. Tällöin alihakemistona olevalla Hakemistolla on luonnollisesti yksi isähakemisto ja hierarkian ylimpänä on yksi juurihakemisto.

Datalla on yksi omistaja, eli Käyttäjä. Käyttäjä voi omistaa Dataa sekä Hakemistoja. Käyttäjän suhde Hakemistoon määritellään Oikeudella. Jos Hakemisto on Käyttäjän A omistama, Käyttäjällä A on oikeudet nähdä ja muokata Hakemistoa ja sen sisältöä. Käyttäjä B voi nähdä ja muokata sitä, jos hänelle on määritetty sellaiset oikeudet. Hakemiston voi asettaa näkymään ja muokattavaksi myös kaikille Käyttäjille.

Käyttäjistä erikoistuu Henkilö sekä Ryhmä. Henkilö on koostumissuhteessa Ryhmään, eli Ryhmät koostuvat Henkilöistä. Ryhmät voivat omistaa Hakemistoja ja Ryhmään kuuluvilla Henkilöillä on oikeuksia näihin Hakemistoihin. Datat ja Hakemistot ovat siis joko Henkilön tai Ryhmän omistamia ja koostumissuhteen avulla Henkilöt voivat Ryhmänsä kautta saada oikeuksia Hakemistojen Dataan. Tällä tavalla ihmiset voidaan jakaa ryhmiin, joilla on omat tiedot näkyvillä vain niiden jäsenilleen.

### 5.3.2 Saantipolkuanalyysi

Saantipolkuanalyysillä arvioitiin sellaisten hakujen onnistuminen, joita arvioitiin käytettävän eniten. Saantipolut on tehty luvussa 3.1.4 esitettyjen yleisimpien käyttötapausten mukaisesti ja niiden toimivuus on todennettu. Tässä esimerkkinä muutama saantipolku ja todennuksen toimivuudesta:

**Käyttötapaus 1 ja 2:** Tietyn henkilön tietyn hakemiston tietty tieto:

$$\text{Henkilö} \longrightarrow \text{Hakemisto} \longrightarrow \text{Data}$$

(\*)

Henkilö-relaatiosta haetaan valittu henkilö. Hakemisto-relaatiosta haetaan valitun henkilön mukaan valittu hakemisto, jonka mukaan haetaan Data-relaatiosta tässä hakemistossa sijaitseva tieto.

**Käyttötapaus 3:** Tiettyyn tietoon liittyvät muut tiedot:

$$\text{Data} \longrightarrow \text{Data}$$

(\*)

Data-relaatiosta haetaan valittu tieto ja sen perusteella kaikki tiedot Data-relaatiosta, joihin ensin haettu tieto liittyy.

**Käyttötapaus 4:** Tiettyyn avainsanaan liittyvät tiedot:

Avainsana — > Data  
(\* )

Avainsana-relaatiosta haetaan valittu avainsana. Data-relaatiosta haetaan valitun avainsanan mukaan kaikki tiedot, joihin valittu avainsana liittyy.

### 5.3.3 Looginen suunnittelu

Tietokannan loogisessa suunnittelussa ensin valittiin käytettävä tietomalli. Valinnassa päädyttiin relaatiomalliin, joka soveltuu yksinkertaisten rakenteiden esitykseen riittävällä tarkkuudella. Suunniteltu käsitteellinen malli on selkeä käsitteistö, jossa staattisilla käsitteillä on selkeät suhteet toisiinsa, eikä monimutkaisia rakenteita löydy. Lisäksi käsitteiden erikoistumiset saa toteutettua helposti relaatiomallilla. Toiminnallisuutta sisältäviä ja tilallisia käsitteitä ei ole, joten oliomallin käyttöä ei nähty tarpeelliseksi. [22]

Käsitteellisen mallinnuksen tuotos käännettiin loogiseksi kuvaukseksi eli relaatiokaavioksi muuttamalla käsitteet suoraan relaatioiksi ominaisuuksineen. Käsitteiden välisiin suhteisiin, joissa lukumääräsuhteet ovat monesta moneen (molemmissa päissä ylärajana on \*), luodaan uusi relaatio kuvaamaan suhdetta ja yhdistämään suhteen päissä olevat käsitteet. Lisäksi näin tehdään tilanteessa, jossa käsitteiden välistä suhdetta kuvaa jo käsite.

Relaatioille määritettiin myös pääavaimet. Pääavainten valinnassa päädyttiin surrogaattien käyttöön, koska minkään relaation ominaisuudet eivät olisi yksilöineet monikoita relaatioissa. Tietokannan käsittekaaviosta (kuva 5.3) käännetty relaatiokaavio on esitetty listauksessa 5.1

Listaus 5.1: Käsitekaavio käännettynä relaatiokaavioksi

**Hakemisto** (@HakemistoID, Nimi, KäyttäjäID->Käyttäjä)  
**HakemistonIsä** (@HakemistoID\_Isä->Hakemisto(HakemistoID), @HakemistoID\_Lapsi->Hakemisto(HakemistoID))  
**Oikeus** (@HakemistoID->Hakemisto, @KäyttäjäID->Käyttäjä, Katseluoikeus, Muutosoikeus)  
**Käyttäjä** (@KäyttäjäID, Nimi)  
**Henkilö** (KäyttäjäID->Käyttäjä, Titteli)  
**Ryhmä** (KäyttäjäID->Käyttäjä, Määritelmä)  
**HenkilönRyhmä** (KäyttäjäID\_Henkilö->Henkilö(KäyttäjäID, KäyttäjäID\_Ryhmä->Ryhmä(KäyttäjäID))  
**Data** (@DataID, Nimi, Luontiaika, Muutos aika, HakemistoID->Hakemisto, KäyttäjäID->Käyttäjä)  
**DatanViittaus** (@Viittaava\_Data->Data(DataID), @Viitattu\_Data->Data(DataID))  
**Avainsana** (@AvainsanaID, Sana)  
**DatanAvainsana** (@DataID->Data, @AvainsadaID->Avainsana)  
**Otsikko** (@DataID->Data, )  
**DatanOtsikko** (@DataID, OtsikkoID)  
**Julkaisu** (DataID->Data, Julkaisijat, Julkaisuvuosi, Julkaisua-lusta)  
**Simulaatio-ohjelma** (DataID->Data, Tekijä, Kohde, Versio)  
**Simulaatiotulos** (DataID->Data, Tekijä, Kohde, Ajoaika, Ohjelmanversio)  
**Kuva** (DataID->Data, Ulottuvuus, Tiedostoformaatti)  
**Muistiinpano** (DataID->Data, Merkkijono)  
**Malli** (DataID->Data, Mallinnuskohde)

## 5.4 XooNIp:s tietokannan ja uudelleen suunnitellun mallin vertailu

### 5.4.1 Käsitteellisten mallien yhtäläisyydet

Kuvan 5.3 uudelleen suunnitellun käsitteellisen mallin ja kuvan 4.1 XooNIps:n tietokannan käsitteellisen mallin peruseriaate on sama. Ryhmiin kuuluvilla käyttäjillä on tietoa, joka on erikoistettu eri tieto-

tyypeiksi. Molemmissa malleissa tieto yksilöidään erikoistamissuhteilla tietyiksi tyypeiksi, eli tallennettava tieto ei voi olla useampaa tyyppiä. Kummassakin mallissa tiedolle on määritetty myös nimi, avainsanat sekä aikaleimat tiedon lisäykselle ja muutokselle.

Uudelleen suunnitellussa mallissa tiedon tyyppiä voidaan määrittää yläluokan tyyppi (Data), jolloin tieto ei ole mikään erikoistetuista tyypeistä. XooNIps:n mallissa tällaista mahdollisuutta ei ole otettu huomioon tiedon erikoistumissuhteiden valinnassa. XooNIps:ssa tietotyyppinä on kuitenkin useita ja File-tyyppistä alaluokkaa voidaan käyttää tyypiltään tuntemattoman tai epämääräisen tiedon tallennukseen.

#### 5.4.2 Käsitteellisten mallien erot

Molemmissa malleissa tieto tallennetaan hakemistoon, mutta rakenteet tähän ratkaisuun ovat poikkeavia. Uudelleen suunnitellussa mallissa hakemisto on erillinen käsite, jossa sijaitsee sen sisältämä tieto. XooNIps:n mallissa hakemisto (indeksi) on yksi tietotyypeistä, joka rekursiokoosteella koostuu tiedosta eli Perus-Itemeistä (tiedostoja kuvaavista itemeistä tai toisista indekseistä). Ratkaisu ei ole paras mahdollinen, sillä indeksissä ja itemeissä ei ole kovin paljon samaa, jolloin Perus-Itemin jotkut ominaisuudet ovat indeksillä turhia.

XooNIps:n mallissa ei oteta myöskään mitenkään huomioon tiedostojen mahdollisia otsikkotiedostoja, joissa määritellään varsinaisen tiedoston tyyppin määrittämiä metatietoja. Otsikkotiedostot on siis tallennettava erikseen järjestelmään ja linkitettävä varsinaiseen tiedostoon käsin, mikä on vaivalloista. Metatietoja kuten itemien ominaisuuksia pystyy muutenkin asettamaan vain rajoitetusti XooNIps:n mallissa. Esimerkiksi kuvatiedostojen ulottuvuustietoja ei voida tallentaa mitenkään, koska Data-tyyppisen itemin tapauksessa voidaan määrittää ainoastaan tiedoston tyyppiä kuvatieosto.

Uudelleen suunnitellussa mallissa otsikkotiedosto on erillinen tietotyyppi (Otsikko) ja suoraan koostesuhteessa varsinaiseen tiedostoonsa, joten yhteys syntyy automaattisesti. Kuvatiedon osalta kuvalle on määritetty ominaisuudet tiedostoformaatille sekä ulottuvuustiedoille.

Muissakin XooNIps:n mallin tietotyypeissä mahdollisuus metatietojen asetukseen on puutteellista verrattuna uudelleen suunniteltuun malliin. Uudelleen suunnitellussa mallissa simulaatio-ohjelma-tyypillä ja simulaatitulos-tyypillä on ohjelman versionumero ja simulaation kohde määritetty toisin kuin XooNIps:n mallissa. XooNIps:n Perus-Item-käsitteellä on ominaisuutena kuvaus, johon voidaan kirjoittaa tekstiä vapaasti, kuten metatietoja. Tämä on kuitenkin huono tapa, koska vapaan tekstin myötä tieto ei ole rakenteista, eli sen merkitys ei ole selvillä ennen kuin tieto on luettu.

## 6. JOHTOPÄÄTÖKSET JA TYÖN ARVIOINTI

Tässä luvussa esitellään johtopäätökset työn tuloksista arvioimalla luvun 5.4 XooNIps:n tietokannan käsitteellisen mallin ja uudelleen suunnitellun käsitteellisen mallin vertailua. Lisäksi luvussa arvioidaan työn tekemistä, sen onnistumista sekä mitä työn tekemisestä opittiin. Luvun lopussa pohditaan yleisellä tasolla vaihtoehtoja XooNIps:lle.

### 6.1 Johtopäätökset

XooNIps täyttää sille asetetut perusvaatimukset, eli tiedon tallennuksen, jakamisen ja hakemisen. Yksityiskohtia tarkasteltaessa siitä paljastuu kuitenkin puutteita tai kehityskohtia, jotka ovat korjattavissa joko sovellusta muokkaamalla tai sen tietokannan rakennetta parantamalla. Tietokannan käsitteellisen mallin uudelleen suunnittelulla pyrittiin löytämään XooNIps:n tietokannan rakenteen kehityskohtiin ratkaisuja, jotta se tarjoaisi vaatimukseen paremmin soveltuvan ja muokattavamman alustan tutkimustiedon tallennukseen.

Suunnittelun tuloksena myös löytyi parannuksia. Tallennettavan tiedon metatietojen muoto ei ole uudessa mallissa yhtä rajoitettua kuin XooNIps:n mallissa. Tutkimusryhmän käyttämien tiedostoformaattien vaatimien otsikkotiedostojen käyttö huomioitiin uudelleen suunnittelussa mallissa. Lisäksi hakemistoihin pohjautuvaa käyttäjien ja ryhmien tallentaman tiedon hierarkkista jaottelua yksinkertaistettiin ja hakemisto käsitteenä erotettiin tieto-käsitteestä.

Hakemistopohjaisen tiedon tallennuksen samankaltaisuus tuli esiin XooNIps:n ja uudelleen suunnitellun mallin vertailussa. Vertailulla pyrittiin todentamaan, että uudelleen suunniteltu malli myös toi jotain uutta XooNIps:n tietokannan rakenteeseen. Vertailu oli onnistunut,

sillä erot saatiin todennettua. Hakemistopohjainen tiedon tallennus tuo kuitenkin esille kysymyksen, olisiko tiedon tallennuksen rakennetta pitänyt muokata enemmän. Hakemistopohjainen tallennus muistuttaa nykyään tavallista tiedostojärjestelmän tapaa esittää tiedostot hierarkkisesti. Tiedon tallennukseen olisi voinut löytyä myös toisenlainen ratkaisu, mutta nyt pysyttiin samassa hakemistopohjaisessa tavassa ja sitä vain muokattiin hieman.

Muutoin vertailulla todennettiin otsikkotiedostojen linkityksen tuoma parannus vaatimusten mukaiseen otsikkotiedostojen tallennukseen sekä parannukset tietotyyppien metatiedon rakenteeseen. Vertailu oli siis onnistunut, mutta kertoi myös uudellen suunnitellun mallinkin tarvitsevan kehittämistä.

## 6.2 Työn arviointi

Neuroinformatiikkaan perehdyttiin tutustumalla erilaisiin alan tieteellisiin julkaisuihin. Tutkimusryhmän jäsenet auttoivat perehtymistä kertomalla alan kehityksestä ja haasteista. Julkaisuihin tutustuminen antoi tarvittavat pohjatiedot työn tekemiseen ja tutkijoiden panos lisäsi näkemystä alaan. Lisäksi syksyllä 2009 vierailu INCF:n vuotuisessa konferenssissa Pilsenissä laajensi ymmärrystä neuroinformatiikan ja neurotieteiden poikkitieteellisyydestä. Kaiken kaikkiaan perehtyminen neuroinformatiikkaa sujui kattavasti ja monipuolisesti.

Tietokannan vaatimuksia pohdittiin tutkimusryhmän jäsenten kanssa. Tässä kohtaa oli tärkeää ymmärtää ero tutkimusryhmän tietokannan käytön sekä yleisen neurotieteiden alan tarpeen välillä. Tutkimusryhmän kohdealue on tietysti rajattu ja siten vaatimukset olivat rajatumpia ja tarkempia. Keskustelujen pohjalta vaatimukset saatiin jäsenneltyä onnistuneesti.

XooNips-sisällönhallintajärjestelmän tutkiminen osoittautui odotettua haastavammaksi. Käyttöliittymän ja sen tarjoamien ominaisuuksien kokeilu oli toki helppoa, mutta XooNips:n käyttämisen tietokan-



nan takaisinmallinnus tuotti aluksi hankaluuksia. Tietokannan dokumentoinnin puute sekä tietokannasta puuttuneet vierasavaimet hankaloittivat takaisinmallinnusta. Ohjelman lähdekoodin tutkimisen sekä XooNips:n kehittäjäryhmän jäseneltä saatujen selvennysten myötä tietokannan rakenne lopulta hahmottui.

XooNips:n tietokannasta suunniteltiin uusi malli, jonka pohjana käytettiin takaisinmallinnettua XooNips:n tietokantaa. Itse suunnittelu sujui onnistuneesti, kun vertalupohjana käytetään XooNips:n tietokannan käsitteellistä mallia. Jälkeen päin ajateltuna nämä työvaiheet olisi voitu tehdä kuitenkin päinvastaisessa järjestyksessä. Jos olisi ensin suunniteltu asetettujen vaatimusten mukainen tietokanta tutkimustiedolle ja vasta sen jälkeen tehty XooNips:n tietokannan takaisinmallinnus, lopputulos olisi voinut olla parempi. Nyt parannellusta versiosta huomaa, että sen suunnittelun pohjana on käytetty XooNips:n tietokantaa ja tämä on alitajuisesti rajoittanut ajatuksia ja ideointia suunnittelussa. Nyt hakemistopohjainen tiedon tallennus ja henkilöiden ja ryhmien suhde tietoon jäivät samankaltaisiksi molempien tietokantojen käsitteellisissä malleissa.

Työn suunnittelussa oli siis kehitettävää, koska parannellun version suunnittelu olisi voinut olla avarakatseisempaa ilman etukäteistä XooNips:iin tutustumista. Lähestymistapa työn ratkaisuun ei ollut aivan selvä työn alkuvaiheessa, jolloin XooNips:ia alettiin jo tutkia. Tässä olisi voinut käyttää harkintaa enemmän ja pohtia ajoissa mitä lähestymistapoja ratkaisuun voisi olla.

XooNips:n käyttöönotto koostui ohjelmiston muokkauksesta sekä ohjelmiston asennuksesta palvelimelle. Muokkaus oli tarpeen, jotta asetetut vaatimukset saatiin täytettyä. Ohjelman lähdekoodiin oli tutustuttava perinpohjaisesti, mutta vasta XooNips:n kehittäjäryhmän jäsenen avustuksella oikeat muutokset saatiin tehtyä. Asennus sujui kohtuullisen hyvin pienten palvelinongelmia lukuunottamatta. Itse käyttöönotto tutkimusryhmän kanssa on kuitenkin vielä tekemättä.

### 6.3 Vaihtoehtoja XooNIps:lle

XOOPS on vain yksi monista avoimen lähdekoodin sisällönhallintasovelluksista, sillä Internetissä on tarjolla kymmeniä muita vaihtoehtoja. Tällaisia ovat esimerkiksi Joomla! [44] ja Drupal [45]. Ne toimivat samaan tapaan kuin XOOPS, eli laajennosta ja moduulien avulla niitä voi muokata vapaasti omaan käyttöön sopivaksi. Kuten XOOPS, nekin on toteutettu PHP:llä ja käyttävät MySQL:ää tietokantana.

Joomla! on useita palkintoja voittanut suosittu sisällönhallintaso-  
vellus, jolla on aktiivinen yhteisö kehittämässä erilaisia laajennoksia ja lisäosia siihen. Joomla!:n käyttö on nopea oppia ja sillä voi luoda kategorioihin perustuvaa sisällönhallintaa. Joomla!:a käyttää eri tarkoituksiin monet tunnetut yritykset kuten eBay, McDonald's, Orange ja Ikea.

Drupal on toinen suosittu sisällönhallintaso-  
vellus ja -kehys, jolla voi toteuttaa monipuolista sisällönhallintaa. Drupalin tuhansilla moduuleilla omaa sisällönhallintaa voi muokata halutunlaisiksi ja sen sanastoiksi nimetyn järjestelmän avulla sisällönhallinnan osia on mahdollista luokitella hierarkkisesti avainsanoilla eri sisältötyypeiksi. Drupal on vaikeampi oppia kuin Joomla!, mutta tarjoaa monipuolisemmat mahdollisuudet sisällönhallinnan toteutukseen. Drupalia käytetään esimerkiksi The Economistin ja Valkoisen talon web-sivustoissa.

Joomla!:lle tai Drupalille ei ole tarjolla vastaavaa moduulia kuin XOOPS:lle on XooNIps. Niinpä nämä vaihtoehtoiset sisällönhallintaso-  
vellukset vaatisivat suunnittelutyötä ennen kuin ne soveltuisivat vaatimusten mukaiseen käyttöön. Toisaalta ne voivat tarjota laajemat ominaisuudet ja sisällönhallinnasta saisi juuri sellaisen kuin itse haluaa.

## 7. YHTEENVETO

Työssä etsittiin ratkaisua TTY:n Signaalinkäsittelyn laitoksella toimivan M<sup>2</sup>oBSI-tutkimusryhmän ongelmaan, miten tallentaa tutkimuksessa syntyvää tietoa hallitusti. Ongelmaa pohdittiin ensin neuroinformatiikan alan kannalta yleisesti. Monimutkainen aivotutkimus vaatii paljon tiedon säilytykseltä ja erityisesti tiedon hallinnalta ja tiedon yhdistäminen on alalla tiedon monimuotoisuuden vuoksi haastavaa. Tutkimusryhmän ratkaisu oli tietokanta, johon tutkimustieto tallennetaan ja käytettäväksi sisällönhallintasovellukseksi valittiin japanilaisten neuroinformatiikan tutkijoiden kehittämä XooNIps.

XooNIps:n ominaisuuksia testattiin koehenkilöiden avustuksella ja tutkittiin, miten se soveltuu tutkimusryhmän vaatimuksiin. Myös XooNIps:n tietokannan rakennetta selvitettiin takaisinmallintamalla se käsitteelliselle tasolle. Käyttökokeiluissa ja vaatimuksia testattaessa kävi ilmi, että XooNIps:ssa on parannettavaa, eikä se täysin vastaa niitä vaatimuksia, joita sille asetettiin. Sovelluksen pahin puute oli tietojen kahdensuuntaisen linkityksen puuttuminen. Tähän pystyttiin tekemään muokkaus, jolla linkitys saatiin toimimaan halutulla tavalla.

XooNIps:n tietokannan käsitteellistä mallia pohjana käyttäen suunniteltiin uusi malli, millä pyrittiin löytämään ratkaisuja XooNIps:n puutteisiin, kuten ennalta määrättyjen tietotyyppien metatietojen tallennuksen rajoittuneisuuteen. Uudelleen suunniteltu malli paikkasi näitä puutteita, joka todettiin vertailemalla XooNIps:n käsitteellistä mallia uudelleen suunniteltuun malliin. Vertailussa tuli myös ilmi, että uudelleen suunnitellussakin mallissa on kehityskohteita, kuten hakemistopohjainen tiedon tallennus. Hakemistopohjainen malli on käytössä

XooNIps:n tietokannassa, eikä sitä lähdetty muuttamaan uudelleen suunniteltuun malliin.

XooNIps:n käyttöönotto jäi kesken, vaikka sovellus saatiin suoritukseen TTY:n Signaalinkäsittelyn laitoksen ylläpidon tarjoamaan web-palveluun. Palvelimella on siis asennettu XooNIps:n muokattu versio, joka ei ole vapaassa levityksessä. Tutkimusryhmän käyttäjien osalta XooNIps:a ei ole vielä otettu käyttöön.

## LÄHTEET

- [1] Beltrame F., Koslow, S.H. Neuroinformatics as a Megascience Issue. *IEEE Transactions on Information Technology in Biomedicine* 3(2001)3, pp. 239-240.
- [2] Business Plan for the International Neuroinformatics Coordinating Facility (INCF) [WWW]. [viitattu 17.11.2009]. Saatavissa: [http://www.incf.org/documents/incf-core-documents/INCF\\_BusinessPlan.pdf](http://www.incf.org/documents/incf-core-documents/INCF_BusinessPlan.pdf)
- [3] Kötter, R. Neuroscience databases: tools for exploring brain structure-function relationships. *Philosophical Transactions of The Royal Society B Biological Sciences* 356(2001)1412, pp. 1111-1114.
- [4] Koslow, S.H., Huerta, M.F. *Neuroinformatics: An Overview of the Human Brain Project*. Mahwah 1997, Lawrence Erlbaum Associates. 382 p.
- [5] Bjaalie, J.G. Annual Report 2006 [WWW]. Tukholma 2007, International Neuroinformatics Coordinating Facility. 12 s. + liit. 4 s. [viitattu 13.11.2009]. Saatavissa: [http://www.incf.org/documents/annual-reports/incf\\_annual2006.pdf](http://www.incf.org/documents/annual-reports/incf_annual2006.pdf).
- [6] INCF Meetings [WWW]. [viitattu 13.11.2009]. Saatavissa: <http://www.incf.org/about/meetings>.
- [7] Neural Systems and Graphics Computing Laboratory [WWW]. [viitattu: 17.3.2010]. Saatavissa: <http://www.nesys.uio.no/>
- [8] Centre for Molecular Biology and Neuroscience [WWW]. [viitattu: 17.3.2010]. Saatavissa: <http://www.cmbn.no/>

- [9] Ascoli, G.A., De Schutter, E., Kennedy, D. An Information Science Infrastructure for Neuroscience Neuroinformatics. Neuroinformatics 1(2003)1, pp. 1-2.
- [10] Beyleveld, D., Townend, D., Rouille-Mirza, S., Wright, J. The Data Protection Directive and medical research across Europe. Aldershot 2004, Ashgate Publishing Limited. 257 p.
- [11] Auranen, T. Computational methods for Bayesian estimation of neuromagnetic sources. Dissertation. Espoo 2007. Teknillinen korkeakoulu, Sähkö- ja tietoliikennetekniikan osasto. Julkaisu - Laskennallisen tekniikan laboratorio. Report B64.
- [12] Phelps, M., Gambhir, S.S., Mahoney, D.K., Markham, J.A. Let's play PET [WWW]. University of California. 1992. [viitattu: 23.11.2009].  
Saatavissa: [http://www.uib.no/med/avd/miapr/arvid/MOD3\\_2002/Bildedannelse/lets\\_play\\_PET.pdf](http://www.uib.no/med/avd/miapr/arvid/MOD3_2002/Bildedannelse/lets_play_PET.pdf)
- [13] Migliore M., Morse T.M., Davison A.P., Marengo L., Shepherd G.M., Hines M.L. ModelDB: Making models publicly accessible to support computational neuroscience. Neuroinformatics 1(2003)1, pp. 135-139.
- [14] Ascoli, G.A., Donohue, D.E., Halavi, M. NeuroMorpho.Org: A Central Resource for Neuronal Morphologies. The Journal of Neuroscience 27(2007)35, pp. 9247-9251.
- [15] Crasto, C.J., Marengo, L.N., Liu, N., Morse, T.M., Cheung, K.H., Lai, P.C., Bahl, G., Masiar, P., Lam, H.Y.K., Lim, E., Chen, H., Nadkarni, P., Migliore, M., Miller, P.L., Shepherd, G.M. SenseLab: new developments in disseminating neuroscience information. Briefings in Bioinformatics 8(2003)3, pp. 150-162.
- [16] SenseLab [WWW]. [viitattu: 27.4.2010].  
Saatavissa: <http://senselab.med.yale.edu/>

- [17] Neuroscience Database Gateway [WWW]. [viitattu: 17.2.2010].  
Saatavissa: [http://www.sfn.org/index.cfm?pagename=NDG\\_main](http://www.sfn.org/index.cfm?pagename=NDG_main)
- [18] Gardner, D., Akil, H., Ascoli, G.A., Bowden, D.M., Bug, W., Donohue, D.E., Goldberg, D.H., Grafstein, B., Grethe, J.S., Grupta, A., Halavi, M., Kennedy, D.N., Marenco, L., Martone, M.E., Miller, P.L., Muller H.M., Robert, A., Shepherd, G.M., Sternberg, P.W., Van Essen, D.C., Williamns, R.W. The Neuroscience Information Framework: A Data and Knowledge Environment for Neuroscience. *Neuroinformatics* 6(2008)3, pp. 149-160.
- [19] Neuroscience Information Framework [WWW]. [viitattu: 17.2.2010].  
Saatavissa: <http://neuinfo.org/>
- [20] Le Novère, N., Bornstein, B., Broicher, A., Courtot, M., Donizelli, M., Dharuri, H., Li, L., Sauro, H.M., Schilstra, M.J., Shapiro, B.E., Snoep, J.L., Hucka, M. BioModels Database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems. *Nucleic Acids Research* 34(2006) Database issue.
- [21] Laibe, Camille. BioModels.net Coordinator, European Bioinformatics Institute. Sähköpostihaastattelu 3.2.2010.
- [22] Connolly, T., Begg, C. Database Systems: A Practical Approach to Design, Implementation and Management. 3. laitos. Harlow 2002, Pearson Education Limited. 1236 p.
- [23] DBMS Architecture [WWW]. [viitattu: 19.1.2010].  
Saatavissa: [http://www.dbmaker.com.tw/reference/manuals/tutorial/tutorial\\_03.html](http://www.dbmaker.com.tw/reference/manuals/tutorial/tutorial_03.html)
- [24] Beaulieu, A. Learning SQL. 2. laitos. Sebastopol 2009, O'Reilly Media. 322 p.

- [25] SQL Tutorial [WWW]. [viitattu: 19.2.2010].  
Saatavissa: <http://www.w3schools.com/sql/default.asp>
- [26] Mullins, C. Database Administration: Practices & Procedures. Boston 2002, Addison-Wesley Professional. 736 p.
- [27] Object Management Group, Inc. Introduction to OMG's Unified Modeling Language (UML) [WWW]. [viitattu: 15.12.2009].  
Saatavissa: [http://www.omg.org/gettingstarted/what\\_is\\_uml.htm](http://www.omg.org/gettingstarted/what_is_uml.htm)
- [28] About MySQL [WWW]. [viitattu: 7.1.2010].  
Saatavissa: <http://www.mysql.com/about/>
- [29] Sun Microsystems Announces Agreement to Acquire MySQL, Developer of the World's Most Popular Open Source Database [WWW]. [viitattu: 7.01.2010].  
Saatavissa: <http://www.sun.com/aboutsun/pr/2008-01/sunflash.20080116.1.xml>
- [30] Oracle and Sun [WWW]. [viitattu: 7.1.2010].  
Saatavissa: <http://www.oracle.com/us/sun/index.htm>
- [31] What is MySQL? [WWW]. [viitattu: 7.1.2010].  
Saatavissa: <http://dev.mysql.com/doc/refman/5.0/en/what-is-mysql.html>
- [32] Chapter 13. Storage Engines [WWW]. [viitattu: 31.3.2010].  
Saatavissa: <http://dev.mysql.com/doc/refman/5.1/en/storage-engines.html>
- [33] Su, C-C. An Open Source Platform for Educators. The 5th IEEE International Conference on Advanced Learning Technologies, Kaohsiung, Taiwan, 5.-10.4, 2005. Washington, DC, 2005, IEEE Computer Society. pp. 961-962.
- [34] Mamba. All About XOOPS [WWW]. [viitattu: 14.4.2010].  
Saatavissa: <http://www.xoops.org/modules/wfchannel/>



- [35] XooNIps User's Manual [WWW]. [viitattu: 23.11.2010].  
Saatavissa: <http://xoonips.sourceforge.jp/manuals/usersman-en/overview.html>
- [36] Yamajia, K., Sakaia, H., Okumuraa, Y., Usui, S. Customizable neuroinformatics database system: XooNIps and its application to the pupil platform 37(2007)7, pp. 1036-1041
- [37] Usui, Shiro. Ph.D, Neuroinformatics Lab, RIKEN Brain Science Institute. Sähköpostikeskustelut 22.5.2009-5.3.2010
- [38] Okumura, Yoshihiro. Neuroinformatics Lab, RIKEN Brain Science Institute. Sähköpostikeskustelut 22.5.2009-5.3.2010
- [39] Methods and Models for Biological Signals and Images [WWW]. [viitattu 12.1.2010].  
Saatavissa: <http://www.cs.tut.fi/sgn/m2obsi/>
- [40] Phelps, M.E., Hoffman, E.J., Huang S.C., Kuhl, D.E. ECAT: a new computerized tomographic imaging system for positron-emitting radiopharmaceuticals. The Journal of Nuclear Medicine 19(1978)6 pp. 635-647.
- [41] Bidgood W.D. Jr, Horii S.C. Introduction to the ACR-NEMA DICOM standard. Radiographics 12(1992)2, pp. 345-355.
- [42] ANALYZE 7.5 File Format [WWW]. [viitattu: 7.4.2010].  
Saatavissa: <http://eeg.sourceforge.net/ANALYZE75.pdf>
- [43] Koskimies, K., Mikkonen, T. Ohjelmistoarkkitehtuurit. Jyväskylä 2005, Talentum Media Oy. 250 p.
- [44] Joomla! [WWW]. [viitattu: 27.11.2011].  
Saatavissa: <http://www.joomla.org/>
- [45] Drupal [WWW]. [viitattu: 27.11.2011].  
Saatavissa: <http://drupal.org/>