

*Juha Kämäräinen & Jarmo Saarti*

# KITSIMU – kirjastojärjestelmän mallintaminen ja ehdotus sen pedagogiseksi sovellukseksi

Juha Kämäräinen & Jarmo Saarti: KITSIMU – kirjastojärjestelmän mallintaminen ja ehdotus sen pedagogiseksi sovellukseksi [KITSIMU – modelling library systems and a proposal for its pedagogical application] *Informatiivitutkimus* 22 (1), 25-30.

This paper is about KITSIMU-project that takes place in the Department of Information Studies, University of Oulu. The aim of the KITSIMU-project is to develop a practical modelling tool for the features of library systems. Object-oriented approaches as well as unified modelling language are used that enable students to participate the modelling process. The pedagogical viewpoint has a great emphasis in the project – there the aim is to give students means to model their own and their patrons' needs for the actual software development projects that take place in the libraries.

*Address: Juha Kämäräinen, Department of Information Studies, P.O. Box 1000 FIN-90014 University of Oulu, e-mail: juha.kamarainen@oulu.fi*

*Jarmo Saarti, Kuopio University Library, P.O.Box 1627, FIN-70211 Kuopio, Finland, e-mail: jarmo.saarti@uku.fi*

## 1. Johdanto

Oli aika, jolloin kirjastojen tietojärjestelmät olivat avoimempia kuin nyt. Kirjaston luettelon teknologiaa edustivat muodoltaan ja kooltaan standardinmukaiset kortit, joihin yhteensopivan hardwaren toteutti puuseppä. Tällaisena järjestelmä oli valmiina kirjastoammattilaisen sisällöllisesti perustaa, käyttää, ylläpitää ja hallita. Nykytilanteessa osin vastaavia tehtäviä suorittavan kirjaston tietojärjestelmän sisäisellä hallinnalla ei voi monikaan kirjastoammattilainen kerskua.

Kirjastoammattilaisen keskeinen työväline on monin osin sulkeutunut hänen kompetenssinsa ulkopuolelle: kaikki kirjasto- ja informaatioaloihin liittyvää työtä tekevät eivät luonnollisestikaan voi olla tietokantojen arkkitehtuureihin ja suunnittelumenetelmiin tai ohjelmointikieliin

perehtyneitä it-ammattilaisia, eivätkä ohjelmistojen lähdekoodit ja tietorakennekuvaukset useinkaan edes ole julkisesti saatavilla. Silti on jotakin tehtävissä sen estämiseksi, että näiden arvokasta informaatiopalvelutehtävää laajan kentän eri laidoilla suorittavien ammattilaisten, järjestelmän tekijän ja käyttäjän, välille revennyt kuuilu ei entisestään levenisi. Tässä artikkelissa kuvattavassa hankkeessa järjestelmän käyttäjä halutaan ymmärtää vähintään potentiaali-sesti aktiiviseksi toimijaksi oman työympäristönsä arvioijana ja osallistuvana kehittäjänä.

KITSIMU (KIrjaston Tietojärjestelmän SIMUlaattori) -hankkeen tarkoituksena on luoda työkalu kirjaston tietojärjestelmien (tästä eteenpäin kirjastojärjestelmä) toimintaperiaatteiden opiskelua, tutkimusta ja kehittämistä varten. Työkalu on järjestelmien toimintaa abstraktilla tasolla simuloiva ohjelmisto.

## 2. Kirjastojärjestelmien kehittyminen

Kirjastojärjestelmä voidaan määritellä tiedonhallintajärjestelmäksi, jonka avulla hallitaan kirjaston kokoelmien käyttö ja tiedonhaku näihin kokoelmiin kuuluvista dokumenteista ja niiden sisällöistä. Kirjastojärjestelmät ovat kehittyneet erilaisista paperiluetteloista ja kortistoista nykyisiksi monipuolisiksi tietokanta- ja tiedonhallintajärjestelmiksi, joissa tiedontallennus ja -haku suoritetaan ohjelmallisilla välineillä ja keinoin. (Saarti 2002, 16.)

Kirjastojen tiedonhallintajärjestelmien kehittämisessä voidaan suomalaisissa yleisissä kirjastoissa nähdä seuraavat järjestelmien sukupolvet (Saarti 2002, 19):

-Ensimmäinen sukupolvi: yksinkertaiset lainausvalvontajärjestelmät.

-Toinen sukupolvi: rajoittuneet tiedonhakumahdollisuudet, käyttöliittymät alkavat kehittyä.

-Kolmas sukupolvi: sisäinen standardimukaisuus (mm. MARC-formaatti, tiedonsiirto), kehittyneet tietokannanhallintajärjestelmät ja relaatiotietokannat, graafisen käyttöliittymäteknikan kehittyminen.

-Neljäs sukupolvi: sekä sisäinen että ulkoinen standardimukaisuus, toimivat rajapinnat sekä tietoliikenteen että muiden sovellusten kesken, modulaarisuus, graafinen, käyttäjiä ohjaava käyttöliittymä, asiakkaan käyttöympäristöstandardiksi muotoutuu Internet, asiakaspalvelinrakenne tietoteknisenä ratkaisuna.

Kirjastojärjestelmät alkoivat siis suomalaisissa yleisissä kirjastoissa kehittyä pitkälti lainauksen hallinnan ongelmista. Suurten kirjastojen logistiikka oli ensimmäinen hallittava ongelma (ks. tästä Linna & Koivisto 1983). Sen jälkeen järjestelmiin tulivat mukaan tiedon tallennukseen ja -hakuun liittyvät osiot. Vastaavasti tiedon tallennus- ja hakujärjestelmien kehitys on ollut erittäin voimakasta koko tietotekniikan kehittymisen ajan. Erillisistä ja yksittäisistä järjestelmistä on siirrytty kohti entistä enemmän integroitua ja tiedonsiirtotekniikoiltaan yhtenäisempiä järjestelmiä. Vaikka suomalaisissa tieteellisissä kirjastoissa kehitys on ollut samankaltaista, siellä on kuljettu kohti askelta tiiviimpää yhteisjärjestelmää. Aluksi toimittiin yhteisen ohjelmiston varassa. Nykyisessä vaiheessa on siirrytty myös kohti yhteisiä tietokantoja ja yhteisiä teknisiä ratkaisuja.

Viime aikoina järjestelmien kehitys on kulkenut kohti digitaalista kirjastoa ja henkilökohtaisia

käyttöliittymiä. Tällöin järjestelmien kautta tulee hallita entistä enemmän sisäisissä ja ulkoisissa verkoissa jaeltavia aineistoja ja tiedon tallennus- ja hakupalveluita (ks. tästä kehityksestä esim. Järvelin 1996, Ingwersen 1999, Kuronen 1996, Talja & Maula 2002). Lisäksi vaatimukset siitä, että kirjastojärjestelmät olisivat myös kirjastojen toimintahallintajärjestelmiä laajassa merkityksessä, ovat kasvaneet.

Hyvän kuvan nykyaikaiseen kirjastojärjestelmään liittyvistä asioista saa Fisherin, Delbridgen ja Lambertin (2001, 344) yhteenvedossa tutkituista 41:stä vaatimusluettelodokumentista. He jakoivat niistä löytyneet toiminnolliset kategoriat seuraaviin ryhmiin:

-järjestelmän asentaminen,

-järjestelmän toiminnot ja ominaisuudet – ydintoiminnot (luettelointi, näyttöluettelo, lainaus, hankinta, johdon tarvitsemat raportit, kaukolainaus, sarjajulkaisu),

-tekniset seikat, mm. laitteisto- ja ohjelmistovaatimukset ja standardit,

-järjestelmän yleiset ominaisuudet, mm. avustustoiminnot, käyttöliittymät,

-huolto ja ylläpito, mm. varmuuskopiointi, ohjelmistopäivitykset,

-toimittajan taustatiedot,

-kirjaston ja sen tietoteknisen ympäristön taustatiedot,

-tuki ja asiakaspalvelu, mm. neuvonta, dokumentaatio ja koulutus.

Tämän kehityksen seurauksena kirjastojärjestelmien hankinnasta ja käyttämisestä on tullut monimutkaisia ja erityisasiantuntijuutta vaativia prosesseja (ks. esim. Stowe 1999). Alan opiskelijoilta ja ammattilaisilta vaaditaan nykyisin pitkälle menevää tietojärjestelmien ja niiden käyttämisen ja mallintamisen hallintaa. Lisäksi oman ongelmansa muodostaa käyttäjien tarpeiden mallintaminen, jotta tietojärjestelmät muuttuisivat kirjastojärjestelmiksi eli kirjastoa ja sen asiakkaiden tarpeita palveleviksi. Eräs olennainen syy tämän alueen hallitsemiseen on myös siinä, että teknologia ja tietojärjestelmät ovat myös tällä hetkellä eräs suuri kirjastojen resurssija sitova tekijä (ks. Saarti 2001).

## 3. Kirjastojärjestelmän tietorakenne ja toimintosimulaattori

KITSIMU:n ensisijaista soveltamisaluetta ovat Fisherin ym. jaottelussa järjestelmän toiminnot

ja ominaisuudet. Houkuttelevaa olisi ajatella simulaattoria käytettävän myös käyttäjien tarpeisiin liittyvään mallintamiseen, mutta tämä alue on epäilemättä teknisen järjestelmän mallintamista huomattavasti hankalampi operationalisoida. Tästä varauksesta huolimatta KITSIMU:sta pyritään tekemään avoin mallintamistyökalu, jonka kaikki sovellusalueet eivät välttämättä ole nähtävissä suunnitteluvaiheessa.

Karkean arkkitehtuurin tasolla KITSIMU koostuu seuraavista osista:

- varsinainen ”simulaattori” eli ohjelmiston se osa, jolla tarkasteltavaa järjestelmää kuvaavat esitykset eli ns. skenaariot suoritetaan
- skenaarioiden laadinta- ja muokkausosa eli editori
- hallintaosa, jolla säädellään simulaattorin ja editorin toimintaan vaikuttavia asetuksia sekä hallitaan järjestelmän tietokantaa
- tietokanta, joka sisältää toisaalta järjestelmän toiminnassa tarvittavat tiedot kuten asetukset ja käyttäjätiedot sekä skenaarioita, jotka tallennetaan editorilla ja ladataan käytettäväksi simulaattorissa

Tietojärjestelmien mallintamisessa voidaan nähdä useita eri tasoja. Asiakkaiden tarpeiden mallintamisessa on hyödynnetty esimerkiksi ns. käyttötapausten (use cases) mallintamista (ks. esim. Cockburn 2002). Tällainen mallintaminen voi tapahtua konkreettisenä kirjoitettuna tekstinä, jossa käyttäjän toiminta jossakin tietojärjestelmän äämpäristössä puretaan askeleittain eteneviksi toiminnoiksi. Tämäntasoiset kuvaukset eli skenaariot muodostavat pohjan itse tietojärjestelmän suunnittelulle ja myöhemmälle ohjelmointityölle. Tärkeää on, että mallintaminen aloitetaan käyttäjän tarpeiden, ei esimerkiksi ohjelmoijien tai teknisen infrastruktuurin lähtökohdista käsin.

Eräs standardiksi muodostunut käyttäjätapausten oliopohjainen mallinnuskieli on UML (unified modeling language), joka lanseerattiin vuonna 1997. Viime aikoina on pelkkien käyttäjien mallintamisen lisäksi alettu korostaa myös sen toimintaympäristön mallintamista, jossa käyttäjien toiminta tapahtuu. Tällöin näkökulma laajenee käyttäjien interaktion ja systeemin suunnittelun näkökulmaksi. UML:n tekee kiinnostavaksi hankkeen kannalta myös se, että kyseessä on useiden eri menetelmien ja notaatioiden sulautuma, jonka taustalta löytyvät oliojattelu merkittävät nimet Booch, Rumbaugh ja Jacobson. Menetelmä on julkinen ja siihen pohjautuvia

välineitä voidaan kehittää ja käyttää ilman korvausta menetelmän kehittäjille. Näin ollen saatavilla on myös lukuisia ilmaisia tai ainakin ilmaiseksi kokeiltavia mallinnustyökaluja. UML sopii siis vähintään löyhästi yhteen KITSIMU:n taustalla jossain määrin vaikuttavan Open source –ideologian kanssa. (Eriksson & Penker 1998, 2000, Open Source.)

Järjestelmän yhteydessä käytetään UML-kuvauskieltä luultavasti kahdella eri tasolla, ellei ensimmäisten selvitysten yhteydessä tule esiin perusteltuja syitä vaihtaa välinettä toiseksi. Nämä tasot ovat itse järjestelmän kehitys ja simulaattorin avulla suoritettavien skenaarioiden kuvaaminen.

Järjestelmän suunnittelun lähtökohdana oleva UML:n komponentti on use case, joka kuvaa järjestelmän toimintaa yksittäisessä tapauksessa sellaisena kuin systeemin ulkopuolinen toimija sen havaitsisi kommunikoidessaan järjestelmän kanssa. Use case –komponentin avulla työstetään siis järjestelmän ”käyttäytymistä”, mutta ei sen sisäistä toiminnallisuutta, jota kuvaavat UML:n muut, luonteeltaan lähempänä toteutusta olevat komponentit. Projektissa näiden soveltamisen aika on myöhemmin, järjestelmän arkkitehtuurin suunnittelusta lähtien. (Eriksson & Penker 1998, 15.)

KITSIMU:n toimintaa ohjaavien skenaarioiden esittämisessä UML on eräs harkittava vaihtoehto. Skenaarioiden kuvaamiseen käytettävällä formalismilla tulee voida kuvata kirjastojärjestelmän sisäisiä tilamuutoksia. UML:n välinevalikoimasta tulee tällöin kysymykseen oliokaavioiden (object diagram) ja tilakaavioiden (state diagram) yhdistelmä, jossa oliokaavio kertoo systeemin sisältämät oliot tietyllä hetkellä ja tilakaavio kuvaa näiden olioiden mahdolliset tilat. Nämä yhdessä tavoittavat kuvattavien kirjastojärjestelmän toimintojen dynamiikan KITSIMU:ssa tarvittavalla tavalla. Simulaattorin luonteeseen kuuluu olennaisesti, että sen toimiessa käyttäjä voi nähdä, mitä kuvattavan järjestelmän sisällä tietyn toiminnon aikana tapahtuu.

KITSIMU:n arkkitehtuuri ja toteutus itsessään noudattaa oliojattelu. Tämä suunnittelun ja ohjelmoinnin paradigman perustuu tietojärjestelmän kohteiden kuvaamiseen olioluokkina ja niiden ilmentyminä. Kullakin luokalla on sille ominainen joukko attribuutteja ja sisäisiä toimintoja. Oliot ja luokat kommunikoivat keskenään lähettämällä toisilleen viestejä. Näin kuvattuna toimiva tietojärjestelmä on siis joukko tiettyihin luokkiin ja hierarkioihin kuuluvia, keskenään viestiviä

objekteja. Toimintamalli on lähellä käyttäytymisen behavioristista ns. ärsyke-reaktio-mallia lukuun ottamatta sitä ajatusta, että suunniteltavien artefaktien kyseessä ollen voidaan niiden sisäpuoli, rakenteet ja toiminnallisuus, toteuttaa halutulla tavalla. Toisaalta tämä merkitsee myös sitä, että mitään piirrettä ei kohteessa ole ennen kuin se on toteutettu.

Olioajattelun keskeisiin periaatteisiin kuuluu ns. periytyminen (inheritance), jonka mukaisesti määritellään keskenään geneerisessä suhteessa olevien luokkien hierarkioita, jotka vastaavat tietojärjestelmän tapahtumien käsittelystä eri tasoilla.

Voidaan esimerkiksi määritellä, että nimekkeet ja niteet ovat luokkana geneerisessä suhteessa toisiinsa (niin oudolta kuin tämä saattaakin kuulostaa). Tällöin niteelle pätee kaikki se, mitä on sanottu nimekkeestä, minkä lisäksi sillä on omia täydentäviä attribuutteja, kuten esimerkiksi signum. Periytystä käyttäen saadaan aikaan helposti muunneltavia rakenteita, joissa yleisemmän tason ominaisuudet ja toiminnot tarvitsee kuvata vain luokkahierarkian ylätasolla.

Jos oliopohjaisen kirjastojärjestelmän käsittehierarkiassa esim. pyydetään tietoa ISBN-numerosta, pyynnön sisältävä viesti etenee hierarkiassa ylöspäin nidetasolta nimeketasolle, jolta pyyntöön reagoiva käsittelijä löytyy (kuvaannollisesti se vastaa: "ISBN-numeroni on..."). Vastaavasti, jos pyyntö kohdistuu tietoon, onko kohde lainassa, sen mielekäs käsittelytaso on nidetaso, jolta saadaan vastaus yksittäistä nidettä edustavalta oliolta toisensa jälkeen ("olen hyllyssä" / "olen lainassa" / "olen kateissa" jne.). Todennäköisesti palvelun pyytjä tarvitsee yhteenvedon tästä kyselykierroksesta. Palvelupyyntö etenee siis hierarkiassa, kunnes sen tunnistava käsittelijä löytyy. Ketjun viimeinen lenkki on virhekäsittelijä, joka antaa ilmoituksen tunnistamattomasta viestistä ("Tämä järjestelmä ei voi käsitellä viestiä X. Valitamme."). Tätä toimintamallia kutsutaan oliotermein monimuotoisuudeksi eli polymorfismiksi: erilaiset oliot kykenevät reagoimaan samaan viestiin, kukin omalla tavallaan.

#### 4. KITSIMU:n koulutussovellukset

Koulutuskäytössä KITSIMU toimii kahdella päätasolla: 1. Kun simulaattoriin on ladattu skenaarioita, niitä askelittain suorittamalla ja rakenteiden tilaa tarkkailemalla opiskellaan koh-

dejärjestelmän toimintaa. Tämä on perustaso, jota voidaan käyttää henkilökohtaisen tai paritöskentelyn lisäksi myös laajemmalle ryhmälle esitettävänä demonstraationa. 2. Skenaarioiden laatiminen sinänsä on tehokas tapa opiskella järjestelmiä muodostamalla malleja niistä ja se voidaan nähdä oppimateriaalin tuottamisen lisäksi myös simulaattorin opiskelukäytön vaativampana tasona. Tällöin toimitaan iteraatiivisesti skenaariota kuvauskielellä rakentaen, testaten ja tarkentaen.

KITSIMU:n käyttö on kumuloituvaa toimintaa: kun simulaattoria käytetään, kohdejärjestelmiä koskevan tietämyksen ohella kasvaa myös niitä kuvaavien skenaarioiden kokoelma. Samalla voidaan luoda skenaariokirjastoja kehittämistyön käyttöön ja toiminta alkaa kumuloitua mahdollistaen erityyppisten ja laajuisten järjestelmien vertailun. Skenaarioiden avulla voidaan myös testata ja mallintaa erityyppisten käyttäjien toimintaa kirjastojärjestelmän parissa – ottaa mukaan rooliajattelu, joka on pedagogisesti tehokas tapa oppia uusia näkökulmia omaan ja toisten toimintaan.

#### 5. KITSIMU:n avulla tehtävä tutkimus ja kehitys

Koska KITSIMU:n käyttö perustuu käsitteelliseen mallintamiseen, voitaisiin väittää, että simulaattorin avulla tehtävä tutkimus tosiasiassa etäännyttääkin tutkimusta todellisuudesta eikä kehitä sitä. KITSIMU:n tutkimussovelluksiin onkin syytä lukea sisään triangulaation periaate: simulaattoritutkimukset on suhteutettava muilla menetelmillä saatavaan tietoon kohdejärjestelmien käyttäjien arjesta (Fabritius 1999, Sawyer 2001). Tutkittaessa uusia ratkaisuja simulaattorimallinnus muodostaa pohjan ohjelmistojen arkkitehtuurin kuvaamiselle sellaisenaan, ennen prototyypin rakentamista tai rinnan sen kanssa käytettynä.

Olellaisena tavoitteena on käyttäjien tarpeiden ja toimintojen analyttinen mallinnus, jonka tulisi aina edeltää tietojärjestelmien laatimista. Viime aikoina kehitys järjestelmissä on kulkenut toisaalta kohti helppokäyttöisiä, ns. tavallisen käyttäjän sovelluksia ja toisaalta kohti analyttisen käyttäjän pitkälle räätälöitävissä ja määriteltävissä olevia sovelluksia. Tiedon tallennus- ja hakusovelluksissa on tarjolla yleensä nämä molemmat vaihto-ehdot. Eräs mielenkiintoinen tutkimuksen kohde on, tarvitaanko ja mihin analyttisiä sovelluksia ja seuraako analyttisistä sovelluksista se, että

käyttäjän osaaminen on suorassa suhteessa sovelluksen monipuolisuuteen ja räätälöityvyyteen. Mitä tukivälineitä sekä aloitteleva että analyttinen käyttäjä tarvitsee järjestelmien käytön tueksi?

## 6. Aikeesta toteutukseen

KITSIMU:n kehittäminen alkaa perusselvityksillä, joihin liittyy mm. käyttökelpoisten mallinnuskielen vaihtoehtojen kartoitus tai kielen vaatimusten määrittely. Varsinaisen simulaattorin ensimmäisiä kuvauksia on ongelman määrittely (problem statement), josta ilmenevät simulaattorin sisäisen maailman eli UoD:n sisältämät entiteetit ja niiden väliset suhteet. Käsite UoD (universe of discourse) viittaa järjestelmän rajojen sisällä olevaan maailmaan, sen sisältämiä kohteita kuvaaviin käsitteisiin, niiden välisiin suhteisiin ja näitä elementteinään käytäviin ilmaisutapoihin. Termi lienee alunperin lainattu järjestelmäsuunnitteluun kielitieteestä (ks. esim. Crystal 1997, 119). Ongelman määrittelyn pohjalta syntyy UoD:n luokkakaavio, jonka lisäksi tarvitaan mm. järjestelmän tiedonhallintaan ja käyttöliittymään liittyviä luokkamäärittelyksiä (Rumbaugh et al. 1991). Rinnan järjestelmän sisäisen tekniikan kanssa aletaan laatia ensimmäisiä skenaarioita, jotka antavat kirjoituspöytätestissä arvokasta tietoa mm. simulaattorin käyttöliittymän suunnittelua ajatellen. Järjestelmä pannaan siis töihin paperilla jo ennen ensimmäisen prototyypin valmistumista. Jo tässä vaiheessa se palvelee opiskelijoiden perehtymistä kirjastojärjestelmien rakenteeseen ja toimintaan.

KITSIMU:n idean ensimmäisiä kenttätestejä on Oulun yliopiston informaatiotutkimuksen laitoksella kevätlukukaudella 2003 järjestettävä aine-opintotasoinen kurssi ”Informaatiojärjestelmien arviointi”, jossa perehdytään ongelmakekseisesti lähestyksen relevanttiin tietojärjestelmäkäsittelyeseen sekä kirjastojärjestelmien arkkitehtuuriin ja toimintoihin. Tavoitteena on antaa osallistujille välineitä järjestelmäympäristönsä kriittiseen arviointiin. Yhtenä työmuotona on järjestelmien piirteiden kuvaaminen käyttäjän havaintojen, kirjallisuuden ja asiakaskäyttöön tarkoitettujen järjestelmädokumenttien perusteella karkean tason pseudokoodia ja UML-kuvauskieltä käyttäen. Näiden harjoitusten myötä saadaan kuvaa siitä, kuinka opiskelijoiden kannalta on tarkoituksenmukaista lähestyä tietojärjestelmien

kuvaamis- ja arvioin-titehtäviä oman alamme näkökulmista kuitenkin hyödyntäen yleisessä käytössä olevia menetelmiä. Toteutuessaan KITSIMU aikanaan toimii näiden kuvausten elävöittäjänä ja analyysivälineenä.

Idean alustavien esittely- ja kehittämissuhteiden jälkeen sille on tehtävä toteutettavuustutkimus, jossa tarkastellaan kriittisesti hankkeen mahdollisuuksia ja riskejä pyrkien ennakoimaan seikat, jotka voivat estää KITSIMU:n järkevän ja realistisen toteuttamisen. Jo tässä vaiheessa voidaan laatia käyttöliittymäsuunnitelmia ja -prototyyppejä sekä järjestelmän käyttöohjeen muodossa kuvausta sen ominaisuuksista eli vaatimusmäärittelyä. Näiden avulla tavoite konkretisoituu. Toteutettavuustutkimuksen tehtävänä on myös tuoda esiin vaihtoehtoja kaavailulle lähestymistavalle sekä selvittää onko sellaisenaan tai muunneltuna tarpeita vastaava väline jo olemassa ja järkevin ehdoin käyttöön otettavissa. Tällöin kokonaan uuden järjestelmän toteuttamisen vaihtoehdoksi tulee olemassa olevan ratkaisun modifikaatio ja soveltaminen. (Pressman 1994, 148, Lowe & Hall 1999, 101, McConnell 1998, 36-43.)

Hyväksytty julkaistavaksi 13.1.2003

## Lähteet

- Cockburn, Alistair (2002). *Writing effective use cases*. 5th pr. (Agile software development series.) Boston, Addison-Wesley.
- Crystal, David 1997. *A dictionary of linguistics and phonetics*, 4th ed. Oxford, Blackwell
- Eriksson, Hans-Erik & Penker, Magnus (2000). *Business modeling with UML: business patterns at work*. New York, John Wiley.
- Eriksson, Hans-Erik & Penker, Magnus (1998). *UML Toolkit*. New York, John Wiley.
- Fabritius, H. (1999). *Triangulation as a multiperspective strategy in a qualitative study of information seeking behaviour of journalists*. In: T. D. Wilson, D. K. Allen (eds.). *Exploring the contexts of information behaviour*. Proceedings of the Second International Conference on Research in Information Needs, Seeking and Use in Different Contexts, Sheffield, UK, 1998. London, UK, and Los Angeles, USA, Taylor Graham Publishing, July 1999. P.406-19
- Fisher, Shelagh & Delbridge, Rachel & Lambert, Siân (2001). *Towards a model system specification for the procurement of library management systems:*

- results of a feasibility study. *Program* 35(4): 339-354.
- Ingwersen, Peter (1999). The role of libraries and librarians in organising digital information. *LIBRI* (49), 11-15.
- Järvelin, Kalervo (1996). Digitaaliset kirjastot tietoverkoissa. *Informaatiotutkimus* 15(2):43-53.
- Kuronen, Timo (1996). Ranganathanin lait ja virtuaalikirjasto. Tampereen yliopisto, Tampere. (Finnish information studies, 4.)
- Linna, Helvi & Koivisto, Tuula (1983). Oulun kaupunginkirjasto: tosiaikainen atk-lainaus käyntiin. *Kirjastolehti* (5):221-222.
- Lowe, David & Hall, Wendy 1999. *Hypermedia & the Web: an Engineering Approach*. John Wiley & Sons.
- McConnell, Steve 1998. Ohjelmistoprojektit: selviytymisopas. Espoo: Suomen atk-kustannus.
- Open Source Initiative OSI - Welcome [The Open Source Home Page]. URL: <http://www.opensource.org/>
- Pressman Roger 1994. *Software engineering: a practitioner's approach*. Third edition, European adaptation. London: McGraw-Hill
- Rumbaugh, James & Blaha, Michael & Premerlani, William & Eddy, Frederik & Lorensen, William (1991). *Object-oriented modeling and design*. Englewood Cliffs (N.J.) : Prentice Hall
- Saarti, Jarmo (2001). Suomen yleisten kirjastojen atk-kirjastojärjestelmät ja niiden kustannusvaikutukset. *Informaatiotutkimus* 20 (2):28-33.
- Saarti, Jarmo (2002). Kirjastojärjestelmät ja niiden toiminnot. Teoksessa: Kirjastojärjestelmän hankkijan opas. Helsinki : BTJ Kirjastopalvelu, 2002. S: 16-42
- Sawyer, Steve (2001). Analysis by Long Walk: Some Approaches to the Synthesis of Multiple Sources of Evidence. In: *Qualitative research in IS : issues and trends*. Hershey : Idea Group. P. 163-189.
- Stowe, Melinda (1999). To RFP or not to RFP: that is the question. *Journal of Library Administration* 26(3/4): 53-74.
- Talja, Sanna & Maula, Hanni (2002). Virtuaalikirjaston rooli tutkijoiden tiedonhankintakäytännöissä. *Informaatiotutkimus* 21(2), 35-50.