# JouTo

## Toimittajan työväline

## Määrittelydokumentti

Versio 1.02

4.3.2001
Helsingin yliopisto
Tietojenkäsittelytieteen laitos
Ohjelmistotuotantoprojekti, kevät 2001, ryhmä 3

| | |
|---|---|
| Projektiryhmä: | Petri Hellemaa (PH) |
| | Antti Koskinen (AK) |
| | Lari Laine (LL) |
| | Jarno Saarto (JS) |
| | Eija Teitto (ET) |
| | Harri Tuuloskoski (HT) |
| Ohjaaja: | Kim Viljanen |
| Vastuuhenkilö: | Raine Kauppinen |
| Asiakkaat: | Helena Ahonen-Myka |
| | Greger Lindén |

## Dokumentin versiohistoria

| | | |
|---|---|---|
| 0.1 | 2.2.2001 | Ensimmäinen versio (LL) |
| 0.7 | 5.2.2001 | Korjattu versio (LL) |
| 1.0 | 8.2.2001 | Valmis dokumentti (LL) |
| 1.01 | 12.2.2001 | Korjattu dokumentti (LL) |
| 1.02 | 4.3.2001 | Korjattu kuvia ja DTD |

Tästä dokumentista vastaa Lari Laine (LL).

# Sisältö

# 1 Johdanto (LL)

Tässä määrittelydokumentissa esitellaan JouTo-ohjelmistotuotantoprojektin tuotteena syntyvä uutistoimittajan työväline. JouTo on lyhenne englannin kielen sanoista *"A Journalist's Tool"*.

## 1.1 Määritelmät, termit, lyhenteet

**CSS** Cascading Style Sheets, W3C:n kehittämä, suosittu HTML- ja XML-dokumenttien ulkoasun määrittelyyn tarkoitettu kieli.
Lisätietoja: `http://www.w3.org/Style/CSS/`

**DoReMi** Document Management, Information Retrieval, Text and Data Mining. TKTL:n tutkimusryhmä, joka vetää TYTTI-projektia.
Lisätietoja: `http://www.cs.helsinki.fi/research/doremi/`

**DTD** Document Type Definition, W3C:n kehittämä rakenteisten dokumenttien rakenteen määrittelykieli.

**EPS** Encapsulated PostScript, kehystetty versio PostScript-kielisestä ohjelmasta. Ohjelmatiedostoon lisätty kehys määrittää rajat tiedoston sisällölle ja mahdollistaa sen sijoittamisen haluttuun kohtaan sivua.

**HTML** Hypertext Markup Language, WWW-sivujen sisällön kuvauksessa käytetty koodaus.
Lisätietoja: `http://www.w3.org/pub/WWW/MarkUp/`

**IPTC** International Press Telecommunications Council, kansainvälinen yhteistyöelin, joka organisoi uutisten määrittelyä ja ryhmittelyä.
Lisätietoja: `http://www.iptc.org/`

**Java** Sunin kehittämä ohjelmointikieli ja järjestelmäriippumaton ajoympäristö.
Lisätietoja: `http://www.sun.com/java/`

**Javadoc** Sunin kehittämä java-koodin dokumentointi kieli
Lisätietoja: `http://www.sun.com/java/`

**Java SDK** Sunin kehittämä Java-ohjelmointiympäristö.
Lisätietoja: `http://www.javasoft.com/products/`

**JDBC** Java Database Connectivity, Java-kielen rajapinta tietokantojen käyttöön.

**NITF** News Industry Text Format, toimittajien kehittämä ja standardoima XML-pohjainen uutisartikkelien määrittelykieli. Lisätietoja: `http://www.nitf.org`

**PostScript** Adoben kehittämä suosittu sivunkuvauskieli. Tulostettava sivu voidaan lähettää PostScript-kirjoittimelle tiiviinä PostScript-ohjelmana tilaavievän raakadatan sijaan.

**TKTL** Helsingin yliopiston tietojenkäsittelytieteen laitos.
Lisätietoja: `http://www.cs.helsinki.fi`

**TYTTI** Tietotyöläisen työasema. Vuosina 2000-2002 käynnissä oleva tutkimusprojekti Helsingin yliopiston tietojenkäsittelytieteen laitoksella, rahoittajina TEKES sekä viisi suomalaisyritystä: Nokia, Alma Media, Sanoma-WSOY, Vaisala ja Lingsoft. TYTTI-projektin vastuuhenkilö Helena Ahonen-Myka ja projektipäällikkö Greger Lindén toimivat JouTo-projektin asiakkaina.
Lisätietoja: `http://www.cs.helsinki.fi/research/doremi/TYTTI/index.html`

**W3C** World Wide Web Consortium, kansainvälinen yhteistyöelin, joka koordinoi WWW-tekniikan ja -standardien kehitystä.
Lisätietoja: `http://www.w3.org/pub/WWW/Consortium/`

**WWW** World-Wide Web, Internetissä toimiva palvelu, joka mahdollistaa tekstin ja kuvien jakamisen helposti ja edullisesti verkon lukemattomille käyttäjille. WWW:ssä aineisto jaetaan sivuihin, jotka sijoitetaan WWW-palvelimelle tavallisina tiedostoina ja hakemistoina.

**XML** Extensible Markup Language, W3C:n kehittämä rakenteisten dokumenttien määrittelykieli.
Lisätietoja: `http://www.w3.org/XML/`

**XSL** Extensible Stylesheet Language, W3C:n kehittämä XML-dokumenttien ulkoasun määritykseen tarkoitettu kieli.
Lisätietoja: `http://www.w3.org/Style/XSL/`

## 1.2 Viitteet

NITFDTD News Industry Text Format DTD version 2.5 8/2000, `http://www.nitf.org`

## 1.3 Dokumentin rakenne

Ensimmäinen luku, Johdanto, esittelee tämän dokumentin rakenteen. Lukuun 1.1 on koottu dokumentissa käytettäviä termejä ja lyhenteitä.

Toisessa luvussa esitellään projekti yleisellä tasolla. Luvussa 2.2 esitellään tuotteen yleiset ominaisuudet ja piirteet. Luvussa 2.3 kuvaillaan tuotteen pääasiallista käyttäjäkuntaa.

Kolmannessa luvussa kuvataan järjestelmän toiminta käyttötapausten avulla. Käyttötapaukset on jaettu kolmeen eri vaiheeseen esiehtoon, kuvaukseen ja lopputulokseen.

Neljännessä luvussa kuvataan järjestelmän arkkitehtuuri ja tiedon kulku tietovuokaavion avulla.

Viides luku kuvaa, järjestelmän rajapinnat kuten käyttöliittymän ja käyttäjälle annettavat palautteet. Tästä luvusta löytyy myös palvelinpuolen kuvaus.

Kuudennessa luvussa eritellään ohjelmiston tietosisältö. Luvussa 6.1 kuvataan ohjelmistoon valittu DTD. Luvussa 6.2 kuvataan tiedon tallentaminen tietokantaan.

Seitsemäs luku luettelee järjestelmän muita ominaisuuksia muunmuassa suorituskykyvaatimuksen ja virheistä toipumisen.

Kahdeksannessa luvussa kerrotaan tuotteen testauksesta. Luvussa käsitellään testausta ensin yleisesti ja sen jälkeen tarkemmin eri tekniikoita, joilla tuotetta tullaan testaamaan.

Luku yhdeksän listaa suunnittelulle ja toteutukselle asetettuja rajoitteita ja standardeja.

Kymmenennessä luvussa täsmennetään tarkemmin jo projektisuunnitelmassa esiteltyä riskienhallintasuunnielmaa.

# 2 Yleiskuvaus (AK)

Tässä luvussa tarkastellaan ohjelmaa yleisellä tasolla. Luvussa esitetään projektin ympäristö (mistä projektissa on kyse), projektin tuotteen toiminta lyhyesti sekä varsinaisen tuotteen käyttäjäryhmien ja käyttöympäristön kuvaukset.

## 2.1 Projektin tarkoitus

JouTo-ohjelmistotuotantoprojekti on keväällä 2001 Helsingin yliopiston tietojenkäsittelytieteen laitoksella järjestettävän Ohjelmistotuotantoprojekti-kurssin näytetyö.

Työ liittyy osittain myös laitoksella käynnissä olevaan DoReMi-tutkimusryhmän "TYTTI- Tietotyöläisen työasema"-projektiin, jossa on mukana rahoittajina joukko suomalaisia teknologiayrityksiä ja Teknologian kehittämiskeskus. JouTo-projektin tarkoituksena ei kuitenkaan ole valmistaa valmista tuotetta kaupalliseen tarkoitukseen. Tuotettava ohjelmisto toimii prototyyppinä, jota voidaan kehittää myöhemmin lisää.

## 2.2 Yleinen toiminta

JouTo-projektin tarkoituksena on tehdä käyttäjälle työkalu, jolla hän voi kätevästi tuottaa artikkeleita. Tuotteen tarkoituksena on luoda käyttäjän selaimeen editori, jonka avulla käyttäjä voi:

- Aloittaa uuden tai ladata vanhan artikkelin
- Editorilla muokata artikkelin metatietoja (otsikko, kirjoittaja yms.) sekä varsinaista tekstiä
- Tallentaa artikkelin

Erilaisille artikkelityypille editori luo erilaisia XML-rakenteita, riippuen kyseisen artikkelin DTD:stä (esim. kuolinilmoitus tai urheilu-uutinen). Editori valvoo myös kyseisen artikkelin DTD-rakenteen oikeellisuutta. Artikkeli tallennetaan tietokantaan, johon artikkeli tallennetaan XML-muodossa.

Kun käyttäjä editoi editorilla artikkelin varsinaista ns. leipätekstiä, hän voi merkata tekstille omia sisältökuvauksia. Näitä sisältökuvauksia ovat mm. (käydään läpi tarkemmin myöhemmin):

- Henkilöt
- Paikkojen nimet (Kaupunki, maa yms)

- Tuotteiden nimet (kirjojen nimet, levyjen nimet yms.)
- WWW- ja email-osoitteiden linkit
- Kuvat, ja niiden yhteydessä olevat kuvatekstit
- Listat ja luettelot
- Lainaukset

Näiden toimintojen lisäksi toimittaja voi määrittää artikkelille otsikon, tiivistelmän, referenssin, julkaisupäivämäärän ja mainita tekijöiden nimet sekä valita uutiselle uutisluokan (Ulkomaat, politiikka jne.).

## 2.3 Käyttäjät ja toimintaympäristö

Ohjelmiston kehitys- ja suunnitteluvaiheessa olemme olettaneet tuotteen editorin käyttäjäkunnaksi eri uutisalueiden toimittajat. Uutistoimittajien tehtäviin nykyisin kuuluu varsinaisen toimittajan työn lisäksi myös artikkelien kirjoitus ja muokkaus sähköiseen muotoon. Tätä työtä varten voidaan käyttää JouTo-projektin tuotetta, jonka käyttäminen ei edellytetä suurta tietoteknistä osaamista toimittajilta.

Oletuksena on myös se, että ohjelmalla tuotettujen artikkeleiden käyttäjäpiiriin kuuluu myös ylläpitohenkilöitä. He voivat hakea toimittajien kirjoittamia artikkeleita tietokannasta ja taittaa ne lehteen sopivaksi sekä suorittaa muita jatkotoimenpiteitä. Ylläpitöhenkilöt voivat myös vastata artikkelien DTD:eistä. Näitä kaikkia toimintoja ei toteuta tuotteemme editorilla vaan niitä voidaan käyttää esimerkiksi tietokannasta käsin.

Uutistoimittajat kirjoittavat artikkeleitaan tietokoneen äärellä usein toimittajan työpaikalla, toimituksessa. Toimintaympäristöksi on oletetty siten toimituksen sisäinen verkko. Toimittajilla on käytössään tietokoneita jotka ovat yhteydessä verkon sisällä olevaan palvelimeen. Artikkelit kirjoitetaan ja muokataan tietokoneella ja tallennetetaan (sekä ladataan) palvelimella olevaan tietokantaan. Koska kyseessä on tomituksen sisäinen verkko, voimme olettaa yhteyksien olevan suojattuja.

# 3 Käyttötapaukset ja toiminnot (PH)

Tässä luvussa on listattu JouTo-järjestelmän käyttötapauksia käyttäjäryhmittäin ja hahmoteltu ohjelmistolle tyypillinen käyttökokonaisuus. Lopuksi on esitetty ohjelmistolta vaadittavat ominaisuudet jaoteltuina toteutettaviin, mahdollisesti toteutettaviin ja ei-toteutettaviin.

## 3.1 Toimittajan käyttötapauksia

Ohjelmisto on suunnattu toimittajille. Se tukee ainakin seuraavanlaisia käyttötapauksia:

Hannu 24 v. kotimaan uutistoimittajana kirjoittaa aamuyöstä jutun eilisestä ullakkotulipalosta Helsingissä Vaasankadulla. Palopäällikkö Veikko Sammuttajan mukaan tilanne oli uhkaava mutta kuolonuhreja ei onneksi ollut.

Koska freelancer-toimittaja Asko 30 v. ei halua kirjoittaa ullakkopalon juttua kokonaan, hän muokkaa Hannun jutusta uuden jutun, oman jutun. Asko lisää yhden kappaleen tekstiä ja laittaa mukaan palopäällikön antaman tilannekuvan.

Kolumnitoimittaja Alias Uuno kirjoittaa valtakunnan suurlehteen kolumnin vedestä käyttäen taustana Pelkosenniemen vesiongelmaa. Lisäksi hän kirjoittaa Alias Aatuna 100 päivää -lehteen artikkelin koirien ulosteongelmasta. Uuno haluaa käyttää koira-artikkelissa kappaletta vesi-artikkelistaan, joka käsitteli kaupungin järjestys- ja puhtaussääntöjä.

Uutistoimittaja Maisa Majakka kirjoittaa ja oikolukee kymmenen artikkelia päivässä. Maisalla on kiire ja artikkelin laatimisen on sujuttava nopeasti ja helposti. Osa artikkeleista on raakavedoksia, osa on editoitavia ja jotkut artikkelit Maisa kirjoittaa itse.

## 3.2 Järjestelmän ylläpitäjän käyttötapauksia

Koska ainakin artikkeleja, niiden rakennemäärityksiä ja mahdollisesti käyttäjäkohtaisia tietoja säilytetään tietokannassa, JouTo-järjestelmän ylläpito olisi lähinnä tietokannan ylläpitoa. Järjestelmän ylläpitäjän käyttötapauksia tukevia ominaisuuksia ei kuitenkaan toteuteta.
Niitä olisivat ainakin:

- käyttöprofiilien lisäys, muokkaus ja poisto
- vanhentuneiden artikkelien tuhoaminen
- dokumenttityyppimäärittelyjen lisäys, muokkaus ja poisto

## 3.3 Tavanomainen käyttökokonaisuus

Ohjelmiston tyypillistä käyttökokonaisuutta on valaistu seuraavalla tilannesarjalla:

1. Toimittaja haluaa syöttää uuden artikkelin JouTo-järjestelmään. Hän käynnistää selaimen ja menee järjestelmän pääsivulle.
2. Toimittaja painaa linkkiä, jolloin editori avautuu uuteen selainikkunaan. Editori-sovelma käynnistyy.

3. Toimittaja valitsee, minkä tyyppisen artikkelin aikoo kirjoittaa ja hänelle generoituu artikkelipohja. Osa artikkelin otsikkotiedoista, kuten kirjoittamispäivämäärä, generoituu itsestään.
4. Toimittaja kirjoittaa artikkelipohjan otsikkotietoihin artikkelin nimen ja valitsee aihepiirin. Sitten hän kirjoittaa leipätekstiä ja liittää sekaan muutaman kuvan.
5. Toimittaja avaa rinnalle toisen editori-ikkunan ja artikkelin. Sieltä hän kopioi muutaman tekstinpätkän uuteen artikkeliinsa ja sulkee sitten ikkunan.
6. Artikkelista on tulossa liian pitkä. Toimittaja kopioi muutaman tärkeän virkkeen kappaleesta toiseen ja poistaa turhat kappaleet.
7. Toimittaja jäsentää vielä artikkelia, valitsee kuvan ja sitä seuraavan kappaleen, ja siirtää ne viimeisiksi.
8. Vahingossa toimittaja tuhoaa yhden otsikon. Hän peruuttaa edellisen toiminnon ohjelman peruutustoiminnolla.
9. Artikkeli alkaa näyttää valmiilta. Toimittaja antaa artikkelille nimen ja tallentaa sen tietokantaan.
10. Toimittaja sulkee editorin ja selaimen.

## 3.4 Toteutettavat toiminnot

Tässä luvussa on lueteltu keskeisimmät ohjelmistolta vaadittavat toiminnot.

### 3.4.1 Editorin käynnistäminen

Artikkeleja muokataan selaimessa toimivassa editorissa. Se täytyy käynnistää.

**Esiehdot:** Käyttäjä on käynnistänyt selaimen ja on JouTo-järjestelmän pääsivulla.

**Kuvaus:** Editori-ikkunan avatakseen käyttäjä painaa linkkiä.

**Lopputulos:** Avautuu uusi selainikkuna. Editori-sovelma latautuu selaimeen ja käynnistyy.

### 3.4.2 Editorin sulkeminen

Toimittaja on artikkelinsa kirjoittanut ja tallentanut ja lopettaa editorin käytön.

**Esiehdot:** Editori on käynnissä.

**Kuvaus:** Käyttäjä sulkee editorin sulkemalla selainikkunan.

**Lopputulos:** Editori on suljettu.

### 3.4.3 Uuden artikkelin luonti

Kiireinen toimittaja haluaa aloittaa uuden artikkelin kirjoittamisen puhtaalta pöydältä. Artikkelin syntymistä nopeuttaa, kun ei tarvitse muistella, mitä kaikkia osia minkäkintyyppiseen artikkeliin piti kirjoittaa.

**Esiehdot:** Ohjelma on käynnissä.

**Kuvaus:** Käyttäjä valitsee uuden artikkelin luonnin. Jos jokin artikkeli on valmiiksi käsittelyssä, hän voi tallentaa sen tässä vaiheessa tai peruuttaa uuden artikkelin luonnin. Samalla käyttäjä valitsee artikkelityypin (DTD:n), jota artikkelin tulee noudattaa.

**Lopputulos:** Valittua DTD:tä noudattava artikkelipohja avautuu editoitavaksi.

Artikkelin voi luoda myös toisen artikkelin pohjalta tallentamalla toisen artikkelin uudella nimellä (katso kohta 3.5.5 Artikkelin tallennus).

### 3.4.4 Artikkelin avaaminen

Toimittaja haluaa jatkaa keskeneräiseksi jäänyttä artikkeliaan tai käyttää kollegansa artikkelia pohjana toiselle artikkelille. Aiemmin tallennettua materiaalia halutaan muokata.

**Esiehdot:** Editori on käynnissä.

**Kuvaus:** Käyttäjä valitsee artikkelin avaamisen. Jos jokin artikkeli on valmiiksi käsittelyssä, hän voi tallentaa sen tässä vaiheessa tai peruuttaa artikkelin avaamisen. Käyttäjä valitsee haluamansa artikkelin.

**Lopputulos:** Artikkeli avautuu muokkaukseen.

### 3.4.5 Artikkelin tallennus

Toimittaja haluaa tallentaa tuotoksensa.

**Esiehdot:** Editorissa on artikkeli auki.

**Kuvaus:** Käyttäjä valitsee joko artikkelin päivityksen (korvaava tallennus) tai tallennuksen uudella nimellä, jolloin hän erikseen hyväksyy mahdollisin muutoksin artikkelin nimen. Jos kyseessä on päivitys, hän voi kirjoittaa siihen liittyvän saatetekstin päiväyksineen. Jos kyseessä ei ole päivitys ja tietokannassa oli ennestään annetun niminen artikkeli, käyttäjän täytyy vahvistaa tai perua tallennustoimenpide.

**Lopputulos:** Artikkeli tallentuu palvelimen tietokantaan korvaten siellä mahdollisesti valmiiksi samalla nimellä olevan artikkelin.

### 3.4.6 Artikkelin sisällön muokkaus

Artikkelin kaikkia osia voidaan muokata rakennemäärittelyn sallimissa puitteissa. Leipäteksti kirjoitetaan elementteihin. Elementeillä voi olla attribuutteja. Attribuutteja käsitellään erillisissä tekstikentissä. Elementtejä voidaan paitsi kopioida ja leikata leikepöydälle myös lisätä, poistaa ja siirtää. Tekstiä voidaan leikata tai kopioida käyttöjärjestelmän leikepöydälle yleisten käytäntöjen mukaan. Vastaavasti leikepöydältä voidaan siirtää tekstiä editoriin. Muokattavaksi valittavan alueen täytyy alkaa ja loppua saman elementin sisällä.

**Esiehdot:** Artikkeli on muokkauksessa.

**Kuvaus:** Käyttäjä kohdistaa muokkaustoiminnon johonkin osaan artikkelia.

**Lopputulos:** Jos rakennemäärittely sallii muokkaustoiminnon, se toteutetaan.

### 3.4.7 Toiminnon peruutus editorissa

Editorissa on yhden askeleen peruutustoiminto, jolla voidaan palauttaa artikkeli tilaan, jossa se oli ennen edellistä muokkaustoimintoa. Myös peruutustoiminto tulkitaan muokkaustoiminnoksi.

**Esiehdot:** Artikkeli, jonka sisältöä on muutettu, on auki editorissa.

**Kuvaus:** Käyttäjä peruuttaa edellisen muutoksen.

**Lopputulos:** Artikkeli on tilassa, jossa se oli ennen edellistä muokkaustoimenpidettä.

### 3.4.8 Usean artikkelin samanaikainen käsittely

Toimittaja haluaa käsitellä useita artikkeleja rinnakkain. Näin voi siirtää materiaalia artikkelista toiseen.

**Esiehdot:** Editori on käynnissä.

**Kuvaus:** Käyttäjä valitsee artikkelin avaamisen (katso edellä kohta 3.5.4 Artikkelin avaaminen) uuteen ikkunaan.

**Lopputulos:** Nykyisen editori-ikkunan rinnalle avautuu toinen editori-ikkuna.

### 3.4.9 Artikkelin katselu XML-muodossa

Joskus on hyödyllistä päästä käsiksi dokumenttiin sen sisäisessä esitysmuodossa. Artikkelin XML-koodin saa näkyviin selainikkunaan.

**Esiehdot:** Editorissa artikkeli auki.

**Kuvaus:** Käyttäjä valitsee artikkelin katselun XML-muodossa.

**Lopputulos:** Avautuu uusi selainikkuna ja sivu, joka sisältää artikkelin XML-koodin.

## 3.5 Mahdollisesti toteutettavat toiminnot

Seuraavia toimintoja toteutetaan, jos se on projektin puitteissa mahdollista. Ne ovat edellä määriteltyihin toimintoihin nähden toissijaisia.

### 3.5.1 Artikkelin poistaminen

Artikkelit vanhenevat. Jos niitä voidaan lisätä tietokantaan, niitä pitäisi voida myös poistaa sieltä.

**Esiehdot:** Editori on käynnissä.

**Kuvaus:** Käyttäjä valitsee artikkelin poiston ja valitsee tuhottavan artikkelin. Käyttäjälle näytetään vielä valitun artikkelin tiedot, minkä jälkeen poistotoimenpide vahvistetaan tai perutaan.

**Lopputulos:** Artikkeli poistetaan tietokannasta.

#### 3.5.2 Artikkelin haku

Artikkeleja voitaisiin hakea artikkelin ja kirjoittajan nimen sekä päiväyksen, aihepiirin, lähteen ja avainsanojen perusteella.

#### 3.5.3 Edistyneemmät leikepöytätoiminnot

Leikepöytätoiminnot voitaisiin ohjelmoida niin, että muutakin kuin tekstiä voidaan siirtää eri sovellusten välillä.

### 3.6 Ei-toteutettavia toimintoja

JouTo-järjestelmään voisi jatkossa lisätä seuraavia toimintoja:

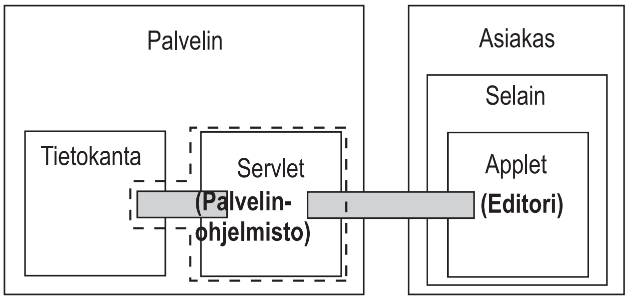
- täydellinen NITF-yhteensopivuus
- käyttäjän autentikointi
- käyttöprofiilit ja niiden ylläpito
- artikkelin lukeminen toisesta muodosta
- artikkelin muuntaminen toiseen muotoon
- artikkelityypin vaihtaminen
- versionhallinta ja monen tason muutoshistoria
- artikkelin käsittely samanaikaisissa istunnoissa
- artikkelin esikatselu HTML-muodossa
- artikkelin tulostus editorista
- liitedatan, kuten kuvien, siirto *drag and drop* -menetelmällä
- artikkelin rakenteen valinta samaksi kuin toisessa artikkelissa
- artikkelin lähettäminen esitaittajalle

## 4 Ohjelmiston arkkitehtuurikuvaus (PH, ET ja HT)

Tässä kappaleessa selvitetään ohjelmiston arkkitehtuuria. Kappale 4.1 sisältää kuvauksen yleisellä tasolla. Kappale 4.2 kuvaa arkkitehtuuria tarkemmalla tasolla tietovuokaavioiden avulla.

## 4.1 Yleinen arkkitehtuurikuvaus

Ohjelmiston arkkitehtuuri on asiakas-palvelin pohjainen. Asiakkaan puolen ohjelmisto on WWW-selaimessa toimiva Java-sovelma (Applet), ja palvelimen puolen ohjelmisto (Servlet) huolehtii tietojen tallennuksesta sekä latauksesta tietokannasta.

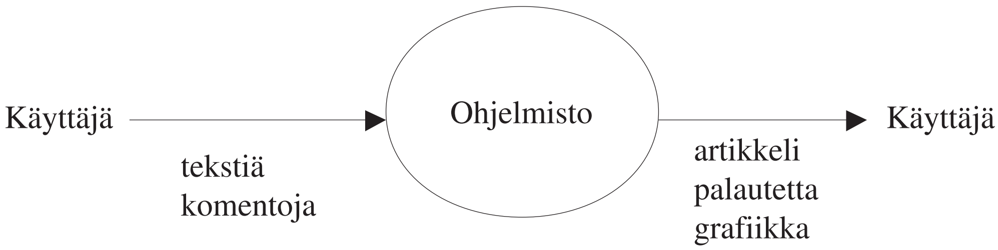



Kuva 1: Ohjelmiston arkkitehtuuri

Kuvassa 1 nähdään ohjelmiston arkkitehtuuri yleisellä tasolla.

## 4.2 Tarkempi arkkitehtuurikuvaus

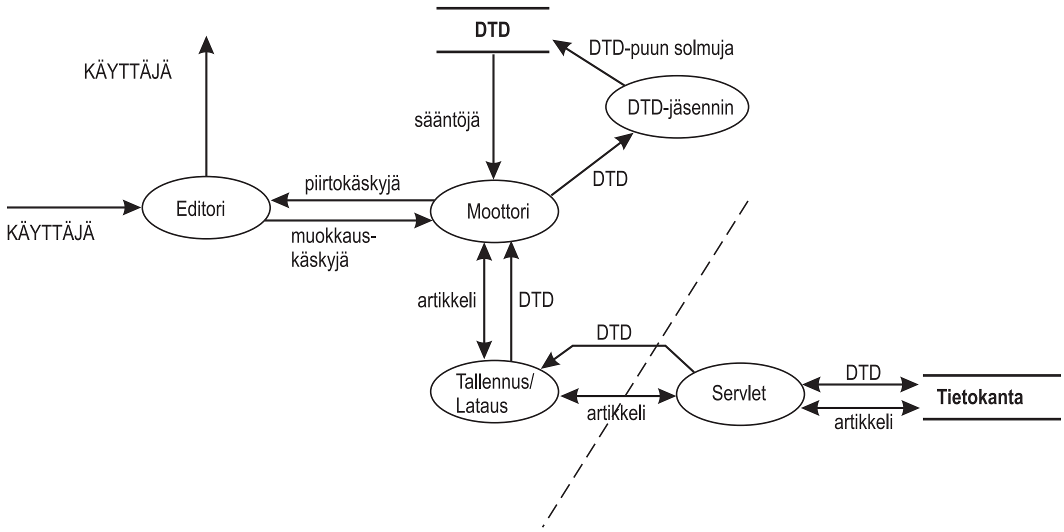
Tässä kappaleessa kuvataan ohjelmiston arkkitehtuuria hieman tarkemmin kuin kappaleessa 4.1.




Kuva 2: Ohjelmiston tietovuokaavio - 1. taso

Kuvassa 2 nähdään ohjelmiston toiminta yleisellä tasolla (tietovuokaavion 1. taso).

Kuvassa 3 nähdään ohjelmiston komponenttien ja erilaisten tietovarastojen sijoittelu ja niiden keskinäinen toimintalogiikka.
Käyttäjä käyttää editoria artikkelien käsittelyyn. Editori on toteutettu Java-sovelmalla (Applet), joka toimii käyttäjän WWW-selaimessa. Käyttäjän ohjelmisto on jakautunut neljään osa-alueeseen, moottoriin, editoriin, DTD-jäsentimeen




Kuva 3: Ohjelmiston tietovuokaavio - 2. taso

sekä tallennus/lataus-osaan.
*Editori-osa* huolehtii käyttöliittymän luomisesta käyttäjälle, sekä käyttäjän syötteiden välittämisestä eteenpäin moottorille. Editori-osa muokkaa käyttöliittymän valitun artikkelin mukaisesti ja huolehtii että käyttäjä voi luoda vain valitun artikkelin mukaisia rakenteita. Editori-osa toimii rajapintana käyttäjän ja varsinaisen ohjelmiston toiminnan välillä.
*DTD-jäsennin* huolehtii valitun DTD:n tulkitsemisesta ja sääntöverkon luomisesta. Tämän verkon avulla valitun artikkelin oikeellisuutta pystytään tarkkailemaan ja käyttöliittymän avulla ohjaamaan.
*Tallennus/lataus-osa* huolehtii kommunikoinnista palvelimen puolen ohjelmiston (Servlet) kanssa. Tallennus/lataus-osan rajapinnan kautta suoritetaan kaikki lataus- ja tallennusoperaatiot.
*Mootori* huolehtii ohjelman toimintalogiikasta ja muiden osien ohjaamisesta. Moottori huolehtii muiden osien keskinäisestä kommunikoinnista sekä suorittaa tarvittavat operaatiot ohjelman ajamiseksi.

Servlet huolehtii artikkelien ja DTD:ien latauksesta, tallennuksesta ja poistamisesta tietokannasta. Servlet on toteuttu Java-sovelluksena, joka pyörii palvelimen Java-moottorin (Java-engine) avulla palvelimen prosessina. Servlet ottaa vastaan editorista tulleita syötteitä, tekee tarvittavat toimenpiteet palvelimen puolella ja palauttaa tulosteet takaisin editorille. Servlet on ainoa komponentti, joka pääsee käsittelemään tietokannan sisältöä. Servlet tarjoaa appletille rajapinnan tietokanta-operaatiohin. Näin tietokannan rakenne pysyy eheänä ja toteutus piilossa ulkopuolisilta elementeiltä.

Tietokantaan on tallennettuna eri DTD:t, DTD:den mukaiset artikkelit, eri käyttäjien tiedot sekä muut JouTo-ohjelmiston tarvitsema informaatio. Tietokanta on

Oracle 8-pohjainen ja sijaitsee Helsingin Yliopiston atk-keskuksen *Kontti*-tietokantapalvelimella. Tietokantaan on luotu tarvittavat taulut ja oikeudet, jotta JouTo-ohjelmistoa pystyy ajamaan.
Tarkempaa tietoa tietokannasta ja sen rakenteesta löytyy kappaleesta 6.2

# 5 Järjestelmän liittymät

Tässä luvussa kerrotaan järjestelmän rajapinnoista ja liittymistä.

## 5.1 Käyttöliittymä

Ohjelmiston käyttöliittymä toteutetaan WWW-selaimeen graafisena java-applettina. Käyttöliittymässä pyritään käyttämään valmiita Java Swing-komponentteja mahdollisuuksien mukaan.
Käyttöliittymä jakaantuu neljään erilliseet osaan, jotka käydään läpi seuraavissa luvuissa.

### 5.1.1 Valikot

Valikoista käyttäjä pääsee ohjaamaan ohjelman niitä osia, joita ei muualta ohjelmasta käsin voi ohjata.

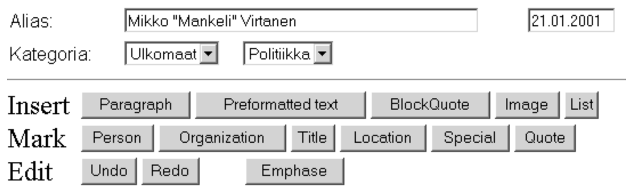
- *File*-valikosta käyttäjä pystyy ohjaamaan artikkelien lataamista, tallennusta, uusien artikkelien luomista sekä vanhojen artikkelien tuhoamista.
- *Edit*-valikosta käyttäjä pystyy ohjaamaan artikkelin ja sen sisällön muokkausta.
- *Option*-valikosta käyttäjä pystyy muokkaamaan editorin ominaisuuksia sekä metadataa.
- *About*-valikosta käyttäjä pystyy näkymään tietoja ohjelmistosta ja sen tekijöistä.


Kuva 4: Editorin valikot

Kuvassa 4 on yksi ehdotelma valikoiden ulkoasusta. Ohjelmiston valikot toteutetaan käytetyn Java-version mahdollisuuksien mukaan.

### 5.1.2 Metatiedot

Metatiedot tarkoittavat varsinaisesta artikkelista suoraan riippumatonta tietoa. Metatietoa voi syöttää editorin valikoiden kautta sekä editorin erilliseen osioon.

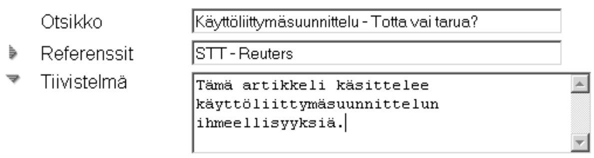



Kuva 5: Metatiedot

Kuvassa 5 nähdään editorin metatiedot-osio. Metatieto-osioon pystyy syöttämään tietoja kirjoittajasta (Alias) sekä artikkelista (Kirjoitushetki ja artikkelin kategoria). Lisäksi metatieto-osiosta löytyy monia muokkaukseen liittyviä toimintoja.

### 5.1.3 Artikkelin pakolliset tiedot

Artikkelin pakolliset tiedot tarkoittavat valittuun artikkelityyppiin liittyviä tiedoja, jotka artikkelin kirjoittajan on ehdottomasti täytettävä ennenkuin artikkeli voidana hyväksyä.




Kuva 6: Artikkelin pakolliset tiedot

Kuvassa 6 nähdään Artikkelin pakolliset tiedot-osiosta löytyvien elementtien esimerkki. Pakolliset elementit ovat erilaisia riippuen valitusta artikkelityypistä, mutta pitää täyttää jotta artikkeli voidaan hyväksyä.

### 5.1.4 Artikkelin valinnaiset tiedot

Artikkelin valinnaiset tiedot tarkoittavat varsinaista artikkelia, eli tekstiä otsikoita, kuvia ja mahdollista muuta artikkelissa olevaa tietoa. Varsinainen artikkeli kirjoitetaan vähän osioon. Tarvittavat elementit, muotoilut ja tyylikeinot saa lisättyä artikkeliin joko valikoista (kappale 5.1.1) tai metatiedot-osiosta (Kappale 5.1.2). Artikkelin kirjoitus tapahtuu JouTo-projektissa tuotettavaan Java-komponenttiin.

Kuvassa 7 näkyy esimerkki artikkelin elementeistä ja kuinka ne näkyvät käyttäjälle.


Kuva 7: Artikkelin valinnaiset tiedot

## 5.2 Ohjelmiston rajapinnat (JS)

Palvelinohjelmiston tulee voida tunnistaa käyttäjä luotettavasti tallennusta ja hakuja varten. Palvelinohjelmisto tarjoaa asiakaspuolelle rajapinnan seuraaviin toimintoihin:

1. Käyttäjän tunnistaminen
2. Olemassa olevien artikkeleiden selaus
3. Artikkelin lataus
4. Artikkelin talletus
5. Artikkelin poistaminen

## 5.3 Syötteet ja tulosteet (JS)

Asiakas- ja palvelinohjelmiston välisen kommunikaation lisäksi palvelinohjelmiston on pystyttävä lukemaan levyltä käytettävä DTD ja tallentamaan asiakasohjelmistolta saamansa tieto tietokantaan.

Käytettävät DTD:t ovat rajauksia NITF-DTD:stä. Tuettavat toiminnot selviävät liitteistä A ja B. Palvelinohjelman tulee kertoa asiakasohjelmalle kulloisenkin DTD:n rajoitukset.

Palvelinohjelma tallentaa asiakasohjelmalla luodut dokumentit Oracle-tietokantaan JDBC rajapintaa käyttäen.

Asiakasohjelmisto antaa käyttäjälle palautetta käyttäjän tekemistä toimista ja mahdollisista virheistä.

# 6 Tietosisältö (ET)

Tietosisältö muodostuu jokaisen artikkelin tiedoista. Artikkelin tietoja käsitellään XML-muodossa. Artikkelin tietojen täytyy olla artikkelille valitun DTD:n mukainen. Varsinainen DTD rakennetaan riittävän laajana (liite A ja B), jotta se kattaa kaikki uutiselle kuuluvat ominaispiirteet. Lisäksi toteutetaan muutama erikois-DTD, jotka ovat varsinaista DTD:tä suppeampia. Tällöin lähinnä eri DTD:den erot ovat tekstiosassa.

## 6.1 Artikkelin osien kuvaus

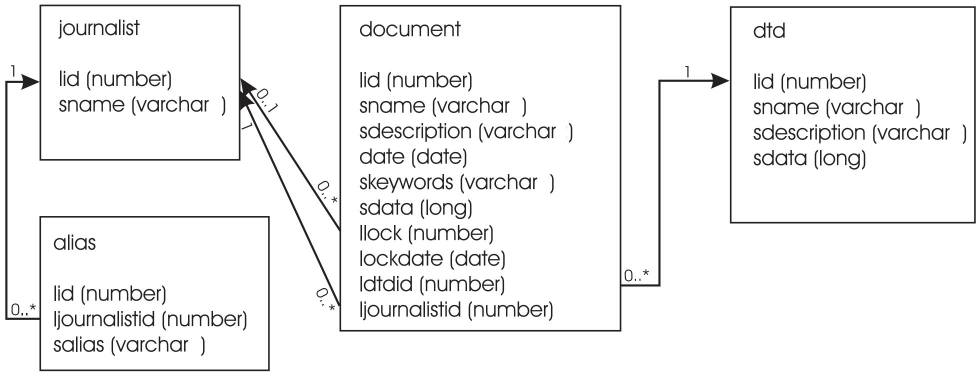
Artikkeli jakautuu kolmeen osaa: hallinnollisiin tietoihin, artikkelin otsikkotietoihin ja tekstiin. Hallinnolliset tiedot ovat järjestelmän tarvitsemia tietoja. Osa hallinnollisista tiedoista rakentuu käyttäjältä piilossa taustalla, esimerkiksi artikkelin kirjoituspäivämäärä. Käyttäjälle näytetään osa hallinnollisista tiedoista, joita käyttävä Artikkeli jakautuu kolmeen osaa: hallinnollisiin tietoihin, artikkelin otsikkotietoihin ja tekstiin. Hallinnolliset tiedot ovat järjestelmän tarvitsemia tietoja. Osa hallinnollisista tiedoista rakentuu käyttäjältä piilossa taustalla, esimerkiksi artikkelin kirjoituspäivämäärä. Käyttäjälle näytetään osa hallinnollisista tiedoista, joita käyttävä voi muuttaa tarvittaessa, esimerkiksi artikkelin julkaisupäivämäärä. Edellä mainittujen tietojen lisäksi artikkelin hallinnollisissa tiedoissa ovat otsikko, kaksitasoinen artikkelin luokitteluhierarkia, kirjoituspaikka, artikkelin id-tieto, kirjoitusoikeuksien omistaja, hakusanat ja versiohistoria. Osa artikkelin tiedoista on pakollisia ja osa vapaaehtoisia. Hallinnolliset tiedot eivät ole julkaistavan artikkelin osia (liite C).

Varsinaisen artikkelin otsikkotiedot rakentuvat pääotsikosta ja mahdollisesti artikkelilla olevasta alaotsikosta. Muita otsikkotason tietoja ovat kirjoittaja tai kirjoittajat, artikkelin alkuperäislähde, tapahtumapäivämäärä ja -paikka ja tiivistelmä. Kaikki artikkelin otsikkotiedot voivat olla julkaistavan artikkelin osia. Jokaiselle artikkelille täytyy antaa pääotsikko, muut artikkelin otsikkotiedot eivät ole pakollisia tai ne ovat valitusta DTD:stä riippuvia.

Artikkelin tekstiosassa voi olla yksi tai monta kappaletta. Kappale voi koostu joko kuvasta ja sen tekstistä tai tekstistä. Artikkelin tekstissä voi olla alaotsikoita, listoja, lainauksia muista teksteistä, henkilön sanojen lainauksia ja artikkeliin liitettyjä kuvia. Editori ei näytä kuvadataa. Lisäksi artikkeli voi sisältää merkattuja paikka-, yritys- ja henkilötietoja ja hyperlinkkejä. Editorin toimintorajoitukset ovat kappleessa 9.2 Ohjelmistorajoitteet.

## 6.2 Tietokannan kuvaus (JS)

Järjestelmään kuluva tietokanta rakentuu neljästä taulusta (kuva 7): *alias, journalist, document ja dtd.* Jokaisessa taulussa on sarake *lid*, joka yksilöi rivin. Tämä




Kuva 8: Taulujen kuvaus

sarake on myös jokaisen taulun pääavain. Tietokantaan kirjoittamisen ja sieltä lukemisen hoitaa yksinomaan palvelinohjelmisto.

### 6.2.1 Taulu alias

*Alias*-tauluun tallennetaan toimittajien mahdolliset nimimerkit eli aliakset. Jokaisella toimittajalla voi olla rajattomasti aliaksia. Ohessa jokainen taulun sarake kuvauksineen.

| **Sarake:** | **Tehtävä:** |
|---|---|
| lid | Pääavain (*number*), joka yksilöi aliaksen. |
| ljournalistid | Vierasavain (*number*), joka viittaa journalist-taulun pääavaimeen (*lid*). Sarakkeen avulla selviää kenen toimittajan alias on kyseessä. |
| salias | Alias tekstinä (*varchar*). |

### 6.2.2 Taulu document

Tauluun *document* tallennetaan editorilla tehdyt dokumentit. Pelkän asiakirjan lisäksi tauluun tallennetaan monia tietoja muun muassa hakujen helpottamiseksi. Ohessa jokainen taulun sarake kuvauksineen.

| **Sarake:** | **Tehtävä:** |
|---|---|
| lid | Pääavain (*number*), joka yksilöi aliaksen. |
| sname | Dokumentin nimi muodossa *varchar*. |
| sdescription | Pidempi selitys dokumentin sisällöstä (*varchar*). |
| date | Dokumentin viimeisin tallennusaika (muodossa *date*). |

| | |
|---|---|
| skeywords | Kuvaavia hakusanoja, jotka helpottavat hakua (*varchar*). |
| sdata | Sarake, johon itse dokumentti on tallennettu (*long*). |
| llock | Kirjoituslukko, joka toimii samalla vierasavaimena journalist -tauluun. Jos lukko on määritelty, näkee samalla kuka sen on varannut. |
| lockdate | Lukon asetusaika (muodossa *date*). Tätä tietoa käytetään varmistamaan, että lukko ei ole jäänyt päälle esimerkiksi tietoliikenneyhteyden katkeamisen vuoksi. |
| ldtdid | Vierasavain, joka osoittaa dtd-taulun pääavaimeen. Sarakkeesta käy ilmi minkä dtd:n mukaan asiakirja on luotu. |
| ljournalistid | Vierasavain (*number*), joka viittaa journalist-taulun pääavaimeen (*lid*). Tämän sarakkeen avulla selviää dokumentin edellinen tallentaja. |
| salias | Alias tekstinä (*varchar*). |

### 6.2.3 Taulu dtd

Tauluun *dtd* tallennetaan dokumenttirakenteen määräävät dtd-tiedostot. Pääavaimen *lid* lisäksi tauluun tallennetaan dtd:n nimi (*sname*), kuvaus (*description*) ja itse dtd, joka tulee sarakkeeseen (*sdata*). Ohessa jokainen taulun sarake kuvauksineen.

| **Sarake:** | **Tehtävä:** |
|---|---|
| lid | Pääavain (*number*), joka yksilöi DTD:n. |
| sname | DTD:n nimi *(varchar)*. |
| sdescription | Pidempi kuvaus DTD:stä *(varchar)*. |
| sdata | Sarake, johon itse DTD on tallennettu (*long*). |

### 6.2.4 Taulu journalist

Tauluun *journalist* tallennetaan toimittajien tiedot. Taulussa on pääavaimen *lid* lisäksi *sname*, johon tallennetaan toimittajan nimi. Ohessa jokainen taulun sarake kuvauksineen.

| **Sarake:** | **Tehtävä:** |
|---|---|
| lid | Pääavain (*number*), joka yksilöi DTD:n. |
| sname | Toimittajan nimi *(varchar)*. |

# 7 Muut ominaisuudet (LL)

Tässä luvussa kerrotaan ohjelmiston muista ominaisuuksista.

## 7.1 Asiakaskoneen suorituskyky (JS)

Asiakaskoneissa pyöritettävä sovellus vaatii koneelta melko vähän. Pentium-teholuokan kone, jossa on mikä tahansa toimiva tietoliikenneyhteys riittää.

## 7.2 Palvelimen suorituskyky (JS)

Laitteistovaatimusten määrittämiseksi on päätettävä sijoitetaanko palvelimen puolen palvelinohjelmisto samalle vai eri koneelle kuin tietokantapalvelin. Taulukossa 1 on kuvattu tarvittava kokoonpano molemmissa tapauksissa. Kovalevytilan määrässä ei ole huomioitu muuta kuin palvelinohjelmiston tarvitsema tila.

| Komponentti | **Kanta eri koneella** | **Kanta samalla koneella** |
|---|---|---|
| Kovalevy | 10 Mt | 510 Mt |
| Muisti | 128 Mt | 512 Mt |
| Prosessori | 300 MHz | 700 MHz |
| Tietoliikenne | 1 MBs | 0.5 MBs |

Taulukko 1 Komponenttien vaatimukset

## 7.3 Ohjelmisto

JouTo-ohjelma ohjelmoidaan Java-ohjelmointikielellä. Ohjelmointiin käytetään Java 2 SDK 1.2.2 -versiota. Sekä palvelimen, että asiakaskoneen *javavirtuaalikoneen* on vastattava toiminnallisuudeltaan tätä versiota. Ohjelmiston toteutus ja testaus suoritetaan *Unix*-ympäristössä. Varsinaisia rajoitteita käyttöjärjestelmällä ei ole, kunhan se täyttää ym. virtuaalikoneen vaatimukset. Asiakaskoneessa selaimen on myös joko tuettava Java 2:sta tai asiakaskoneeseen on asennettava erillinen lisäohjelma, jolla tuki saadaan lisättyä selaimeen.

Palvelinohjelmisto testataan *Oracle 8.0.5* -tietokannalla. Toteutuksessa ei kuitenkaan käytetä mitään Oracle-ympäristöstä riippuvia työkaluja, joten se saattaa toimia hyvin myös muiden tietokantaversioiden kanssa. Palvelimelle on määriteltävä käyttöoikeudet niin, että palvelinohjelmisto pääsee käyttämään JouTo-projektille varattuja tietokannan tietorakenteita. Jos palvelimen puolen ohjelmistoa suoritetaan toisella palvelimella kun varsinainen tietokanta sijaitsee on tietokantayhteyden oltava niin tehokas, että vasteajan pidentyminen ei aiheuta haittavaikutuksia palvelinohjelmiston toimintaan.

*Tiedon varmistus* on hoidettava niin, että kaikki tiedot ovat palautettavissa takaisin esimerkiksi levyrikon sattuessa. Tiedon varmistuksesta vastaa tietokannan ylläpito. Levyrikon tai muun virhetilanteen tapahtuessa pitää vuorokauden ja sitä vanhemmat tiedot pystyä palauttamaan.

## 7.4 Virhetilanteet ja niistä toipuminen

Palvelin- ja asiakasohjelmiston on pystyttävä tunnistamaan omat virhetilanteensa ja toipumaan niistä luotettavasti. Palvelin- ja asiakasohjelmiston on pystyttävä informoimaan asiakkaita omista ja asiakkaiden aiheuttamista virheistä.

Laitteistopohjaisten tai force majeure -tyyppisiin virhetilanteisiin ei palvelin- tai asiakasohjelmiston puolella varauduta.

## 7.5 Ylläpidettävyys

Projektissa tuotettavat dokumentit tallennetaan Postscript-muodossa käyttäen yhteistä mallia. Lähdekoodit kirjoitetaan yhteisen tyylioppaan mukaisesti. Dokumentit ja lähdekoodit tallennetaan projektin kotihakemistoon, jossa ne ovat tarvittaessa helposti saatavissa. Dokumentit löytyvät projektin kotisivuilta osoitteesta http://www.cs.helsinki.fi/group/jouto ja lähdekoodit sijaitsevat tietojenkäsittelytieteen laitoksen tiedostopalvelijan hakemistossa /home/group/jouto/k01/src. Ohjelmiston ylläpidettävyyttä parantaa myös artikkeleiden ja DTD-dokumenttien tallentaminen tietokantaan, jolloin ne ovat helposti käsiteltävissä.

# 8 Testaus (HT)

Tässä kappaleessa kerrotaan ohjelmiston testauksen yleispiirteet, käytettävät testausprotokollat, testauksen vaiheistuksesta sekä henkilöstöstä.

## 8.1 Yleistä

Testauksen tarkoituksena on löytää ohjelmistosta mahdollisimman suuri osa virheistä ennen ohjelmiston julkaisua. Testaus on valitussa projektimallissa oma vaihe, joka suoritetaan varsinaisen toteutuksen jälkeen. Testausvaiheessa kuitenkin

oletetaan, että toteutusvaiheesa jokainen ohjelmoija on esitestannut omaa ohjelmakoodiaan, jottei testausvaiheessa tarvitsisi puuttua aivan alkeellisimpiin ohjelmointivirheisiin.
Ennen varsinaista testausvaihetta laaditaan testaussuunnitelma. Testaussuunnitelmaan kirjoitetaan tarvittavat testausprotokollat ja jaetaan tehtävät. Testausdokumentti kirjoitetaan varsinaisen testausvaiheen aikana, ja siihen kirjataan löydetyt virheet ja tehdyt korjaukset.
Testausvaiheen vastaava on testausvaiheen projektipäällikkö. Hän vastaa että testausvaihe suoritetaan asianmukaisesti riittävän laajana ja kattavana. Projektipäälikkö vastaa myös testausvaiheen tulosten kirjaamisesta testausdokumenttiin.

## 8.2 Testausmenetelmät

Testauksessa käytetään neljää (4) eri testausmenetelmää, jotka testaavat ohjelmiston eri osa-alueita: toiminnallisuutta, ohjelmakoodin tasoa, käyttöliittymän käytettävyyttä sekä valmiin ohjelmiston aistinvaraista toimivuutta.

### 8.2.1 White-Box -testaus

White-Box - testauksessa pyritään löytämään ohjelmakoodista joitain puutteellisuuksia tai toimintalogiikan virheitä. Jokainen ohjelmoija tarkkailee aktiivisesti jo toteutusvaiheessa omaa ohjelmakoodiaan. Varsinaisessa White-Box - testausvaiheessa testataan tärkeimmät algoritmit sekä kriittisin osa-alue ohjelmakoodista.

### 8.2.2 Black-Box -testaus

Black-Box-testauksessa pyritään löytämään virheitä ohjelmiston rajapinnoista ja komponenttien välisestä toiminnasta. Black-Box testaukseen osallistuvat kaikki tuotetut komponentit.

### 8.2.3 Käyttöliittymätestaus

Käyttöliittymätestauksessa pyritään etsimään toimimattomia tai huonosti toimivia piirteitä ohjelmiston käyttöliittymästä ja korjaamaan ne. Käyttöliittymätestausvaiheessa testataan asiakaspuolen (Client) ohjelmisto.

### 8.2.4 Palvelinpuolen testaus

Palvelinpuoli testataan sekä White-Box että Black-Box - testauksella. White-Box testaukseen osallistuvat ohjelmiston toiminnan kannalta kriittiset komponentit. Black-Box - testauksellta testataan koko ohjelmiston toiminta testaussuunnitelmassa määritellyin protokollin.

### 8.2.5 Valmiin ohjelmiston testaus

Valmiin ohjelmiston testaus kokonaisuutena testaa ohjelmiston toimivuutta normaalikäytössä. Testauksella on tarkoitus löytää vikoja tai ongelmia ohjelmistossa tai jossain sen osassa, mitä ei muulla testauksella on pystytty havaitsemaan.

## 8.3 Testauksen suorittajat

Testausvaiheesta vastaa vaiheen projektipäällikkö. Testaukseen varaudutaan jo suunnittelu- ja totetusvaiheessa tekemällä kattava testaussuunnitelma, josta käy ilmi suunnitelmat testauksen suosittamiseksi määrätyssä aikataulussa.
Testaus suoritetaan testaussuunnitelmassa todetuin protokollin, toimenpitein ja projektipäällikön määräämin henkilöresurssein. Testausvaiheessa löydettyjen virheiden korjaajiksi määrätään joko alkuperäisen ohjelmakoodin kirjoittaja tai joku muu projektin jäsen projektipäällikön päätöksien mukaan.
Testausvaihe katsotaan loppuneeksi, kun testaussuunnitelmassa olevat kohdat on testattu ja niistä löydetyt ohjelmavirheet on korjattu. Lopputulos katselmoidaan koko projektin voimin, ja testauksen loppumisesta ja tuloksesta päätetään yhteisesti.

# 9 Rajoitteet suunnittelulle ja toteutukselle (LL)

Tässä kappaleessa kerrotaan mitä rajoitteita ohjelmiston suunnittelulle ja toteutukselle on asetettu.

## 9.1 Noudatettavat standardit

Lähdekoodissa käytetään Java 1.2.2:ssa määriteltyjä ominaisuuksia.

Ohjelmakoodi dokumentoidaan käyttäen Javadoc-standardia.

Artikkelit noudattavat määriteltyä DTD:tä (liite A ja B), joka noudattaa NITF-DTD:n versiota 2.5 [NITF]. Artikkelit ja NITF-DTD ovat XML-standardin mukaisia.

## 9.2 Ohjelmistorajoitteet

Järjestelmässä toteutetaan vain valitun DTD:n sisältö (liite A ja B). Tekstiosuutta muokattaessa ei sallita rekursiivisiä sisäkkäisiä lohkorakenteita, ts. kappaleen sisälle ei voi sijoittaa toista kappaletta ja lista ei voi sisältää toisia listoja. Lohkorakenteita ovat mm. otsikot, kappaleet, kuvat ja listat.

# A Liite: DTD-elementit

Seuraavassa taulukossa esitetään JouTo-projektissa käytettävät NITF-elementit ja niiden attribuutit.

| Pääelementti | Elementti | Attribuutti |
|---|---|---|
| NIFT | | BASELANG |
| | | |
| HEAD | TITLE | |
| | TOBJECT | kaikki elementit |
| | | Subject 2-tasoisena |
| | DOCDATA | Evloc |
| | | Doc-id |
| | | Doc.copyright |
| | | Date.issue |
| | | Keylist.keyword |
| | | |
| BODY | | |
| BODY.HEAD | HEDLINE | Hl1 |
| | | Hl2 |
| | BYLINE | |
| | DISTRIBUTOR | |
| | DATELINE | Story.date |
| | | Story.location (1-tasoinen) |
| | ABSTRACT | |
| | | |
| BODY.CONTENT | P | Location (1-tasoinen) |
| | | Org (1-tasoinen) |
| | | Person (1-tasoinen) |
| | | Br |
| | | Em |
| | | Q |
| | | Object.title |
| | OL | |
| | UL | |
| | PRE | |
| | MEDIA | Media-type="image" |
| | | Id |
| | | media-caption: enriched text |
| | | media-reference: data-location |

# B Liite: Kommentoitu DTD-dokumentti

```
 <!--
JouTo Document Type Definition
http://www.cs.helsinki.fi/group/jouto/

Project Group JouTo - Helsinki University - Department Of Computer Science - 2001

This DTD is defines the structure of documents used by JouTo project.
This DTD is a part of NITF-DTD version 2.5
http://www.nitf.org/
-->


<!-- ***************************************************************
DTD Entities
*************************************************************** -->

<!--    ***************************************************************

        DTD Entities

        *************************************************************** -->



<!ENTITY % enriched-text '
        #PCDAT
        | location
        | object.title
        | org
        | person
        | a
        | br
        | em
        | q
        '>

<!ENTITY % block.content "p | hl2 | media | ol | ul | pre ">

<!-- Attribute entities -->

<!ENTITY % global-attributes '
        id      ID                                              #IMPLIED
        '>
<!ENTITY % url.link '
        md      CDATA                                           #IMPLIED
        '>

<!--    ***************************************************************
        nitf
        *************************************************************** -->

<!ELEMENT nitf (head, body)>
<!ATTLIST nitf
        %global-attributes;
        uno             CDATA           #IMPLIED
        version         CDATA           #FIXED          "-//IPTC-NAA//DTD NITF-XML 2.1//EN"
        change.date     CDATA           #FIXED          "4 July 2000"
        change.time     CDATA           #FIXED          "1900"
        baselang        CDATA           #IMPLIED
        class           NMTOKENS        #IMPLIED
        >
        <!--
```

```
        Element                 nitf
        Partial description     The root element for NITF.
        Full description        News Industry Text Format instance. Holds head and body.

        Attribute name          id
        Description             An XML-specific identifier for the element.
        Attribute name          uno
        Description             The universally unique identifier for the document.
        Attribute name          version
        Description             This DTD's formal public identifier.
        Attribute name          change.date
        Description             Date when this DTD was last changed.
        Attribute name          change.time
        Description             Time when this DTD was last changed.
        Attribute name          baselang
        Description             Base language for document. Uses en-US (per ISO 639 and 3166).
        Attribute name          class
        Description             An open placeholder for categorization.
        -->

<!--    ****************************************************************
        head
        **************************************************************** -->

<!ELEMENT head (title?, tobject?, docdata?)>
        <!--
        Element                 head
        Partial description     Holds metadata about the document as a whole.
        Full description        Generally does not hold information for direct display to user.
Portions of the metadata, though, are often rendered to augment an article.
        -->

<!ELEMENT title (#PCDATA)>
        <!--
        Element                 title
        Partial description     Document Title.
        Full description        Often displayed within a result-list after a search finds this article.
        -->

<!ELEMENT tobject (tobject.subject)>
<!ATTLIST tobject
        tobject.type            CDATA   "news"
        >
        <!--
        Element                 tobject
        Partial description     Subject code.
        Full description        Used to identify the type of the news material, based on a Subject Code system.
        -->

        <!ELEMENT tobject.subject  EMPTY>
        <!ATTLIST tobject.subject
                tobject.subject.ipr             CDATA           "IPTC"
                tobject.subject.refnum          NMTOKEN         #REQUIRED
                tobject.subject.code            CDATA           #IMPLIED
                tobject.subject.type            CDATA           #IMPLIED
                tobject.subject.matter          CDATA           #IMPLIED
                >
                <!--
                Element                 tobject.subject
                Partial description     Assigns subject information to news material based on a Subject
Code system.
                Full description        A three-tiered hierarchy consisting of broad subjects, secondary topics
(Subject Matter), and tertiary topics (Subject Detail).

                Attribute name          tobject.subject.ipr
                Description             Information provider reference.
```

```
                Attribute name          tobject.subject.refnum
                Description             Eight-digit reference number for all three levels.
                Attribute name          tobject.subject.code
                Description             Three-letter code for the subject type.
                Attribute name          tobject.subject.type
                Description             First-tier subject description.
                Attribute name          tobject.subject.matter
                Description             Second-tier subject description.
                -->

<!ELEMENT docdata (
        evloc?,
        doc-id,
        date.issue,
        doc.copyright?,
        key-list?
        )>
        <!--
        Element                 docdata
        Partial description     Document metadata.
        Full description        Container for metadata information about this particular document.
        -->

        <!ELEMENT evloc EMPTY>
        <!ATTLIST evloc
                county-dist     CDATA   #IMPLIED
                city            CDATA   #IMPLIED
                >
                <!--
                Element                 evloc
                Partial description     Event location.
                Full description        Where an event took place (as opposed to where the story was written).

                Attribute name          county-dist
                Description             County or district name.
                Attribute name          city
                Description             City or municipality name.
                -->

        <!ELEMENT doc-id EMPTY>
        <!ATTLIST doc-id
                id-string       CDATA   #IMPLIED
                >
                <!--
                Element                 doc-id
                Partial description     Registered identification for document.
                Full description        Character string that provides a unique, persistent identification
for a document.

                Attribute name          id-string
                Description             Document identification string.
                -->

        <!ELEMENT date.issue EMPTY>
        <!ATTLIST date.issue
                norm            CDATA   #IMPLIED
                >
                <!--
                Element                 date.issue
                Partial description     Date/time document was issued.
                Full description        If not provided, it is assumed to be on receipt.

                Attribute name          norm
                Description             Date/time value normalized to ISO 8601 (YYYYMMDDThhmmssZ),
normalized to UTC.
                -->
```

```
	<!ELEMENT doc.copyright EMPTY>
	<!ATTLIST doc.copyright
		year		NMTOKEN		#IMPLIED
		holder		CDATA		#IMPLIED
		<!--
		Element			doc.copyright
		Partial description	Copyright information for document header.
		Full description	A metadata location for the information about the year and holder of
the document copyright. This information should be consistant with
information in the copyrite tag.

		Attribute name		year
		Description		Year of the document copyright.
		Attribute name		holder
		Description		Holder of the copyright.
		-->

	<!ELEMENT key-list (keyword)*>
		<!--
		Element			key-list
		Partial description	List of keywords
		Full description	hold a list of keywords about the document.
		-->

		<!ELEMENT keyword EMPTY>
		<!ATTLIST keyword
			key		CDATA	#IMPLIED
			>
			<!--
			Element			keyword
			Partial description	Keyword or phrase.
			Full description	Holds the keyword or phrase itself.

			Attribute name		key
			Description		The actual keyword.
			-->


<!--	****************************************************************
	body
	**************************************************************** -->

<!ELEMENT body (body.head?, body.content*)>
	<!--
	Element			body
	Partial description	The content portion of the NITF document.
	Full description	All items within body are meant for display to the user.
	-->

<!--	****************************************************************
	body.head
	**************************************************************** -->

<!ELEMENT body.head (
	hedline?,
	byline*,
	distributor?,
	dateline*,
	abstract?,
	)>
	<!--
	Element			body.head
	Partial description	Metadata intended to be displayed to the reader.
	Full description	Holds content such as headline, byline, etc.
	-->
```

```
<!ELEMENT hedline (hl1, hl2*)>
        <!--
        Element                 hedline
        Partial description     Container for main headline and subheadlines.
        Full description        Provides a generally-dependable component for display in search results.
        -->

        <!ELEMENT hl1 (%enriched-text;)*>
        <!--
        Element                 hl1
        Partial description     Headline 1 (main-headline).
        Full description        The primary headline of the story.
        -->

        <!ELEMENT hl2 (%enriched-text;)*>
        <!--
        Element                 hl2
        Partial description     Headline 2 (sub-headline)
        Full description        A subordinate headline for the article.
        -->

<!ELEMENT byline (#PCDATA | person | location)*>
        <!--
        Element                 byline
        Partial description     Container for byline information.
        Full description        Can either be structured with direct specification of the responsible person /
 entity and their title or unstructured text can be provided.
        -->

<!ELEMENT distributor (#PCDATA | org)*>
        <!--
        Element                 distributor
        Partial description     Information distributor.
        Full description        May or may not be the owner or creator.
        -->

<!ELEMENT dateline (#PCDATA | location | story.date)*>
        <!--
        Element                 dateline
        Partial description     Container for dateline information.
        Full description        The date and location where the story was created.
        -->

        <!ELEMENT story.date (#PCDATA)>
        <!ATTLIST story.date
                norm            CDATA   #IMPLIED
                >
                <!--
                Element                 story.date
                Partial description     Date of story.
                Full description        Printable date, with normalized version as attribute.

                Attribute name          norm
                Description             Normalized date and time.
                -->

<!ELEMENT abstract (%block.content;)+>
        <!--
        Element                 abstract
        Partial description     Story abstract
        Full description        summary or synopsis of the contents of the document.
        -->

<!--    ****************************************************************
        body.content
```

```
        ****************************************************************** -->

<!ELEMENT body.content (%block.content;)*>
        <!--
        Element                 body.content
        Partial description     Actual body content.
        Full description        Holds paragraphs, lists, tables, media, and other containers.
        -->

<!ELEMENT p (%enriched-text;)*>
        <!-

        Element                 p
        Partial description     Paragraph.
        Full description        An organized set of sentences.
        -->

<!-- hl2 is a block.content member -->

<!ELEMENT media (media-reference, media-caption?)>
<!ATTLIST media
        media-type      (
                        image
                        )       #REQUIRED
        >
        <!--
        Element                 media
        Partial description     Generalized media object.
        Full description        Used to package up a reference to an image, audio, video, etc. with
its caption and credit.

        Attribute name          media-type
        Description             Audio or video or photo etc. (Controlled vocabulary.)
        -->

        <!ELEMENT media-reference (#PCDATA)>
        <!ATTLIST media-reference
                data-location           CDATA           #REQUIRED
 mime-type  CDATA   #REQUIRED
                >
                <!--
                Element                 media-reference
                Partial description     Reference to a media object.
                Full description        Points to any object, such as photo, audio, video and text;
or to executable files.

                Attribute name          data-location
                Description             Location of data other than a URN, such as a absolute server path.
      Attribute name mime-type
 Description Mime-type for the external media file.
                -->

        <!ELEMENT media-caption (%enriched-text;)*>
                <!--
                Element                 media-caption
                Partial description     Text describing media.
                Full description        Often rendered beneath a photo or video.
                -->

<!ELEMENT ol (li)+>
        <!--
        Element                 ol
        Partial description     Ordered list.
        Full description        Holds list items.
        -->
```

```
<!ELEMENT ul (li)+>
        <!--
        Element                 ul
        Partial description     Unordered list.
        Full description        Holds list items.
        -->

        <!ELEMENT li (%enriched-text;)*>
                <!--
                Element                 li
                Partial description     List item.
                Full description        Holds content that is presented within a list.
                -->

<!ELEMENT pre (#PCDATA)>
        <!--
        Element                 pre
        Partial description     Preformatted information.
        Full description        Carriage returns and fixed-width spacing should be maintained.
        -->

<!--    ****************************************************************
        text elements
                location | object.title | org | person | a | br | em | q
        **************************************************************** -->

<!-- copyrite allowed here, also -->

<!ELEMENT location (#PCDATA)>
        <!--
        Element                 location
        Partial description     Significant place mentioned in an article.
        Full description        Used to normalize locations.
        -->

<!ELEMENT object.title (#PCDATA)>
<!ATTLIST object.title
idsrc  CDATA  #REQUIRED
value  CDATA  #REQUIRED
>
        <!--
        Element                 object.title
        Partial description     Title of inline object such as book, song, artwork, etc.
        Full description        Used to normalize these values.

       Attribute name idsrc
  Description The source (taxonomy) for the element's value.
  Attribute name value
  Description The code or symbol for the element.

        -->

<!ELEMENT org (#PCDATA)>
<!ATTLIST org
idsrc  CDATA  #REQUIRED
value  CDATA  #REQUIRED
>

        <!--
        Element                 org
        Partial description     Organization.
        Full description        Used for companies (both private and public) and non-profits.

       Attribute name idsrc
  Description The source (taxonomy) for the element's value.
  Attribute name value
```

```
  Description The code or symbol for the element.
        -->

<!ELEMENT person (#PCDATA)>
<!ATTLIST person
idsrc  CDATA  #REQUIRED
value  CDATA  #REQUIRED
>

        <!--
        Element                 person
        Partial description     Human individual.
        Full description        Used to normalize people.

       Attribute name idsrc
  Description The source (taxonomy) for the element's value.
  Attribute name value
  Description The code or symbol for the element.
        -->

<!ELEMENT a (%enriched-text;)*>
<!ATTLIST a
        href            CDATA           #IMPLIED
        name            CDATA           #IMPLIED
        %url.link;
        title           CDATA           #IMPLIED
        >
        <!--
        Element                 a
        Partial description     Anchor for hypertext links.
        Full description        Allows users to traverse to other documents.

        Attribute name          href
        Description             URL for where the hypertext anchor leads.
        Attribute name          name
        Description             Alternate name or description of the link, such as "Link to news item."
        Attribute name          title
        Description             Title for the document to be linked to.
        -->

<!ELEMENT br EMPTY>
        <!--
        Element                 br
        Partial description     Forced line break.
        Full description        Often used within headlines.
        -->

<!ELEMENT em (%enriched-text;)*>
        <!--
        Element                 em
        Partial description     Emphasis.
        Full description        Can be rendered as bold, italics, or another style.
        -->

<!ELEMENT q (%enriched-text;)*>
<!ATTLIST q
        quote-source    CDATA   #IMPLIED
        >
        <!--
        Element                 q
        Partial description     Quotation.
        Full description        Words directly attributed to an individual.

        Attribute name          quote-source
        Description             Who said or wrote the quotation.
        -->
```

# C Liite: Artikkeliesimerkki (ET)

Helsingin Sanomien uutista 'Kriittinen lukutaito tarpeen äidinkielen opetuksessa' on käytetty pohjana artikkeliesimerkissä, uutinen julkaistiin 29.1.2001. Alkuperäiseen uutiseen verrattuna artikkeliin on lisätty tekstiä ja merkkauksia ja suurin osa alkuperäisestä tekstistä jätetty pois.


```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE nitf SYSTEM "JOUTO.DTD">
<nitf>
<head>
  <title></title>
  <tobject>
   <tobject.subject
    tobject.subject.ipr="a455"
    tobject.subject.refnum="11.22.33"
    tobject.subject.code="knu"
    tobject.subject.code="kotimaa"
    tobject.subject.matter="uutinen"
    />
  </tobject>
  <docdata>
   <evloc="Helsinki"/>
   <doc-id="3000.15.a"/>
   <date.issue norm="2001-01-25"/>
   <date.release norm="2001-01-29"/>
   <doc.copyright="STT"/>
   <keylist>
      <keyword key="lukutaito"/>
      <keyword key="opetus"/>
      <keyword key="teksti työnä, virka kielenä"/>
  </keylist>
  </docdata>
</head>
<body>

<body.head>
   <hedline>
    <hl1>Kriittinen lukutaito tarpeen äidinkielen opetuksessa</hl1>
    <hl2><q>Teksteissä rutkasti arvoja ja asenteita</q></hl2>
   </hedline>
   <byline>Tiina Vilkas</byline>
   <distributor>SST</distributor>
   <dateline>
      <story.date norm=" 2001-01-20"/>
      <story.location="Helsinki"/>
   </dateline>
   <abstract><p>Äidinkielen opetuksessa painopisteen tulisi kohdistua
kriittiseen lukutaitoon. Vuorovaikuttamisen näkökulmasta on mietittävä
millaisia rooleja kirjoittajalle ja lukijalle tuotetaan.
   </p></abstract>
</body.head>

<body.content>
<p><location>Oulu</location>. Äidinkielen opetus kaipaa enemmän
kriittiseen lukutaitoon perehdyttämistä. <org>Kotimaisten kielten
tutkimukeskuksen</org> erikoistutkijan <person>Veikko Heikkisen</person>
mukaan koululaisille tulisi opettaa tekstin tarkastelua muutenkin kuin vain
tiedonvälittäjinä, sillä teksteistä on luettavissa rutkasti arvoja ja
asenteita.
</p>
<media media_type="image">
    <media-reference data-location="opettajat.gif"/>
```

```
    <media-caption><q>Äidinkielen opettajien valtakunnallisen
kokoontumisen osanottajia<q></media-refence>
</media>
<p>Heikinen luonnehtii lainaten tutkimuksensa <object.title>Teksti
    työnä, virka kielenä</object.title>
  <q><em>Sisällön tarkastelu hahmottaa, millainen maailma tekstillä
rakennetaan:</em> onko siinä esimerkiksi paljon toimintaa ja tapahtumia
vaiko asioden suhteita tai käyttäytymisen kuvaamista</q>
  tekstinsisällön merkitystä ratkaisevaksi.
</p>
    <hl2>Sivistyksellinen kahtiajako uhkaa</hl2>
<p><location>Oulu</location>.Äidinkielen päiville muita puhujia
olivat seuraavat henkilöt:
</p>
<ol><li><person>tutkija Pirjo Hiidenmaa</person></li>
    <li><person>tutkija Ulla Tiililä</person></li>
    <li><em>kirjailija<special>Leena Krohn</special></em></li>
</ol>

</body.content>
</body>
</nitf>
```