

Toteutusdokumentti

Orava

Helsinki 5.5.2005

Ohjelmistotuotantoprojekti

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

Kurssi

581260 Ohjelmistotuotantoprojekti (6 ov)

Projektiryhmä

Juhani Bergström

Peter von Etter

Teppo Känsälä

Olli Lyytinen

Jessika Penttinen

Mikko Waris

Asiakas

Eero Hyvönen

Johtoryhmä

Juha Taina

Hannu Räisänen

Kotisivu

<http://www.cs.helsinki.fi/group/orava/>

Versiohistoria

Versio	Päiväys	Tehdyt muutokset
0.1	21.1.2005	Runko
1.0	5.5.2005	Palautus

Sisältö

1 Johdanto	1
1.1 Terminologia	1
1.2 Dokumentin rakenne	2
2 Xml2rdf	2
3 Ontologiat	3
3.1 Oppiaine	4
3.2 Kohderyhmä	4
3.3 Teema	4
3.4 Taso	4
3.5 Rooli	4
3.6 Ohjelmatyyppi	5
3.7 Ohjelmasarja	5
3.8 Ohjelma	6
3.9 Teemalabels	6
4 Metatiedot	6
4.1 Klaffin videot	6
4.2 MuseoSuomen kohteet	7
5 Päätelysäännöt	7
5.1 Kategoriasäännöt	7
5.1.1 Oppiaine	8
5.1.2 Teema	8
5.1.3 Ohjelmasarja	8
5.1.4 Ohjelmatyyppi	8
5.1.5 Kohderyhmä	8
5.1.6 Rooli	8
5.1.7 Vaativuus	8
5.1.8 Oppimisympäristö	9
5.1.9 Mediatyyppi	9
5.2 Suositteelusäännöt	9

	ii
5.2.1	Sama aihe 9
5.2.2	Sama osa sarjassa 9
5.2.3	Seuraava osa sarjassa 9
5.2.4	Edellinen osa sarjassa 10
5.2.5	Sama aihe MuseoSuomessa 10
5.2.6	Sama paikka MuseoSuomessa 10
6	Käyttöliittymä 10
6.1	XSLT-muunnokset 10
6.1.1	oravaXhtml.xslt 10
6.1.2	oravaXhtmlItem.xslt 11
6.1.3	oravaItems.xslt 12
6.1.4	oravaGroup.xslt 12
6.1.5	oravaLogo.xslt 12
6.1.6	oravaUrl.xslt 12
6.2	CSS-tyylit 13
6.3	Käyttöliittymän toteutus 13
6.3.1	Yleisosa 13
6.3.2	Etusivu 13
6.3.3	Valintasivu 13
6.3.4	Kohdesivu 16
6.3.5	Avainsanahaku 16
6.3.6	Ohje 18
7	Ontoviews muutokset 18
7.1	Ustategen.xsp 18
7.2	OrvTransform 19
8	Asennuskriptit 19
9	Vaatimukset 20
10	Tunnetut ohjelmavirheet 21
Lähteet	21

1 Johdanto

Tämä on Ohjelmistotuotantoprojekti-kurssin Orava-ryhmän tuottaman Orava-portaalin toteutusdokumentti. Portaali on semanttisen webin [SemWeb] tekniikoilla toteutettu haku- ja suosittelukone YLE Opetusohjelmien Klaffi-videoarkistoon [Klaffi], jonka videot ovat nykyisin verkossa käytettävissä Opinportti-palvelun [Opinportti] kautta. Portaali on rakennettu soveltaen MuseoSuomi-hankkeessa [MuseoSuomi] kehitettyä OntoViews-järjestelmää [OntoViews], jolla videot saadaan linkitettyä semanttisesti toisiinsa.

1.1 Terminologia

Ontologia Määrittelee käsitteitä ja niiden välisiä suhteita. Koko järjestelmässä ontologioita on useita: esimerkiksi paikkaontologia joka sisältää eri paikkojen määritelmät, ja aikaontologia joka määrittelee eri aikakäsitteitä.

(Ontologinen) käsite Ontologiassa yksiselitteisesti määritelty käsite. Käsitteitä on kahdenlaisia: luokkia ja yksilöitä. Luokkakäsite voi sisältää yksilökäsitteitä ja luokkia. Yksilökäsite taas viittaa johonkin yksilöolioon. Esimerkiksi yksilökäsite ”vuosi 2005” kuuluu luokkakäsitteeseen ”2000-luku”.

Kohde Yksilökäsite, joka viittaa tiettyyn tietokannan videoon tai muuhun kohteeseen.

Suosittelukohde Käyttäjän parhaillaan tarkastelemaan kohteeseen semanttisesti liittyvä kohde, jota portaali suosittelee käyttäjälle nykyisen kohteen yhteydessä.

Kategoria Kohteiden järjestämiseen käytetty käsite.

Näkymä Hierarkisiksi taksonomioiksi järjestetyt kategoriat.

Valinta Käyttäjän tekemä rajaus näytettäviin kohteisiin. Käyttäjä voi lisätä rajoituksia valitsemalla näkymistä kategorioita tai lisäämällä hakusanoja.

Moninäkökymä-käyttöliittymä Käyttäjän valintojen mukaan näytettävä näkymien joukko.

LOM Lyhenne engl. sanoista *Learning Object Metadata*. Tietomalli opetukseen liittyvien objektien metadatalle. [LOM]

XML Lyhenne engl. sanoista *Extensible markup language*. Yleinen rakenteellinen kuvauskieli. [XML]

XSL Lyhenne engl. sanoista *Extensible stylesheet language*. XML-muotoinen kieli, jolla muunnetaan XML-dokumentteja eri muotoisiksi XML-dokumenteiksi. [XSL]

RDF Lyhenne engl. sanoista *Resource Description Framework*. Semanttisen webin tietomalli metadatalle. [RDF]

RDFS Lyhenne engl. sanoista *Resource Description Framework Schema*. Ontologioiden määrittelyyn käytetty RDF-tietomalli. [RDFS]

URI Lyhenne engl. sanoista *Universal resource identifier*. Yksilöllinen tunniste jollekin resurssille, mihin täytyy voida viitata esimerkiksi RDF-dokumentissa. [URI]

OntoViews MuseoSuomi-portaalin ohjelmisto. [OntoViews] Koostuu osajärjestelmistä Ontogator ja Ontodella.

(Prolog-)päätelysääntö Prolog-kielillä ilmaistu sääntö, jota käytetään kategorioiden tekemisessä ja suositeltujen kohteiden hakemisessa.

Metadata Yleisesti, tietoon liittyvää tietoa. Tässä dokumentissa tarkoittaa Klaffi-arkiston videoihin liittyvää tietoa videoista, esimerkiksi videon teknisiä tietoja, sisältöön liittyviä avainsanoja jne. Joissakin kohdissa tarkoitetaan myös MuseoSuomen metadataa, joka ei liity videoihin vaan MuseoSuomen materiaaliin. Sama metadata esiintyy järjestelmässä erimuodoissa eri komponentteja varten. Ylläpitodokumentissa [YD] on selitetty tarkemmin eri komponenttien metadatumuodot.

1.2 Dokumentin rakenne

Luvut 2 – 8 kuvaavat miten Orava-portaaliin kuuluvat komponentit toteutettiin. Luvussa 9 selvitetään miten toteutettu ohjelmisto täyttää sille asetetut vaatimukset ja luvussa 10 kuvataan minkälaisia virheitä ohjelmisto tiedetään sisältävän.

2 Xml2rdf

XML2RDF on Java työkalu, joka muuntaa XML-dokumentteja RDF-muotoon. Se toteutettiin kuten suunnitteludokumentissa on kuvattu. Joitakin luokkia tehtiin lisää suunnitelmaan verrattuna, jotta muunnos saatiin toimimaan hyvin. Nämä käyvät ilmi XML2RDF:n javadoc-dokumentaatiosta, joka on liitteenä ylläpitodokumentaatiossa [YD].

Orava-projektissa Klaffin lähdetiedostot olivat LOM-tietomallin mukaisessa XML muodossa, mutta muunnosprosessi suunniteltiin siten, että sitä voidaan käyttää myös muun mallisille XML-tiedostoille. XML2RDF:n muokkaus ja ylläpito on kuvattu tarkemmin ylläpitodokumentissa [YD].

XML2RDF koostuu eristäjäluokista (Java-luokka *InfoExtractor*), joista kukin hakee tietoa lähdetiedostoista. Kukin eristäjä hakee lähteestä jonkin tietyn tiedonpalasen, esimerkiksi videon nimen tai videon avainsanat, tai prosessoi haettua tietoa jollakin tavalla. Eristäjiä liitetään yhteen siten, että toisen tulos toimii toisen syötteenä. Näin muodostetut eristäjäketjut voivat tehdä monimutkaisiakin operaatioita vaikka jokainen eristäjäluokka itsessään on hyvin yksinkertainen. Eristäjät välittävät tietoa toisilleen kokoelma luokissa (*java.util.Collection*), jolloin on mahdollista erotella syötteestä useita erikseen käsiteltäviä osia. Eristäjät käyttävät Java version 1.5:n generisyys-ominaisuutta tyyppimääritys-

ten tarkastamiseen käänösvaiheessa. Tämän takaa sen, että eristäjiä ei voi yhdistää sillä tavalla, että toisen tulos ei sovi syötteeksi toiselle.

Lähdetiedostot ladataan yksi kerrallaan DOM-muistitietorakenteeseen. Kukin näistä käsitellään *CollectedInfoExtractor* -luokalla, joka käyttää kokoelmaa eristäjiä tiedon hakemiseen. Ensiksi tuotetaan RDF-resurssi yhdellä eristäjäketjulla. Tämän jälkeen tähän resurssiin lisätään ominaisuuksia muiden eristäjäketjujen avulla. *CollectedInfoExtractor* -luokassa käytetään kuvausta (*java.util.Map*), jolla kuvataan RDF-ominaisuuksia eristäjäketjuille. Tämä kuvaus määrittelee ominaisuudet, joita resurssiin liitetään ja eristäjät, jotka tuottavat arvot niille. Kukin eristäjä tuottaa tuloksena kokoelman joko RDF-resursseja tai literaali-arvoja. Kunkin eristäjäketjun syöteenä taas toimii XML-tiedostoa vastaava DOM-elementti.

Kun muunnetaan XML-tiedostoja, ketjun ensimmäinen eristäjä saa syötteeksi itse XML:n. Orava-projektissa ensimmäinen eristäjä hakee tietoa XPath-tekniikalla (Java-luokka *XPathInfoExtractor*). Tämä tuottaa XML-dokumentista poimitun merkkijonon, jota jatkokäsitellään muilla eristäjillä esimerkiksi pilkkomalla se osiin pilkkujen kohdalta tai suorittamalla haku ja korvaus -operaatioita säännöllisillä lausekkeilla. Lopuksi eristetty arvo muutetaan RDF-resurssiksi tai literaaliksi. Resurssin tapauksessa tässä vaiheessa eristetty arvo otetaan talteen ja sitä käytetään pian luotavan resurssin nimenä. Nimestä tehdään URI, jonka avulla voidaan tehdä resurssi. Tämä tehdään luokassa *TermCollectorExtractor* .

Projektin aikana huomattiin, että kaikki metadatan lähdetiedostot eivät ole täysin XML-formaatin mukaisia. Tämän takia XML2RDF:ään liitettiin työkalu, joka korjaa tiedostot mahdollisimman hyvin. Lisäksi huomattiin, että metadatatassa esiintyy joitakin tiedostopareja, joissa kumpikin tiedosto sisältää saman URL-osoitteen videolle. Koska ei pystytä päättämään kumpi näistä on virheellinen, päätettiin toimia niin, että molemmat videot jätetään pois tuotettavasta RDF tiedostosta ja ruudulle tulostetaan varoitus, jotta ylläpitäjä voi käsin korjata virheen.

3 Ontologiat

Eräs tavoitteista oli, että järjestelmään voidaan lisätä mahdollisimman helposti uusia vastaavia videoita. Tämän vuoksi muunnos Klaffin metadatatista järjestelmän mukaiseen RDF-muotoon on automatisoitu mahdollisimman paljon.

Ontologiat ohjelma, ohjelmasarja ja teemalabels tuotetaan xml2rdf-työkalun muunnoksen tuloksena automaattisesti. Muut ontologiat on luotu käsin.

Suunnitteludokumentista [SD] poiketen oppimisympäristö-ontologiasta luovuttiin. Lisäksi asiapitoisuutta ja vaikeusastetta kuvataan kumpaakiin taso-ontologialla.

3.1 Oppiaine

Oppiaineet ontologia määrittää hierarkisen puun, jonka jäseninä erilaiset oppiaineet ovat. Oppiaineet voivat kuulua johonkin yleisempään opintoainetta kuvaavaan luokkaan. Esimerkiksi *Historia* on oppiaineet-ontologian käsite ja kuuluu luokkaan "*Humanistiset tiedet*", mistä seuraa hierarkinen rakenne.

Metadatatassa vastaava tieto on esitetty siten, että oppiaine esiintyy avainsanana.

3.2 Kohderyhmä

Kohderyhmä-ontologia kuvaa kohdevideon käyttäjäryhmiä. LOM-metadasta tämä tieto löytyy kentästä *lom/educational/typicalAgeRange*.

Kohderyhmällä ominaisuutena ovat *aloitusika* ja *lopetusika*, jotka kuvaavat kiinnostuneen ryhmän tavanomaista ikää. Tämä mahdollistaa kohderyhmien käytön iän ja logiikan sääntöjen avulla.

3.3 Teema

Teema-ontologian ensisijaisena lähteenä toimivat LOM-metadatan avainsanat.

Uniikkeja avainsanoja on 3860. Mikäli lasketaan vain avainsanat, jotka esiintyvät vähintään kolme kertaa putoaa uniikkien avainsanojen määrä alle tuhanteen. Avainsanoja esiintyy kaiken kaikkiaan 13621. Sata yleisintä avainsanaa esiintyy 4466 kertaa. Eli noin kolmasosa avainsanoista kuuluu näiden sadan yleisimmän avainsanan joukkoon. Useat näistä yleisimmistä avainsanoista ovat oppiaineita ja joukossa on lisäksi samasta kantasanasta eri kirjoitusasuja. Kun vielä huomataan, että suuri osa avainsanoista esiintyy vain kerran tai pari, niin hyödyllisten hakua raajaavien avainsanojen joukko on hyvin rajallinen.

Teema-ontologian käsitteet on valittu siten, että ne tarjoaisivat järkevän kokonaisuuden oppijoille, joiden kiinnostukset voivat olla hyvinkin erilaisia. Käytännössä avainsanojen rajallisuus saneli jo yksinään minkälaisia käsitteitä voidaan järkevästi käyttää.

3.4 Taso

Taso-ontologia kuvaa jonkin asian vaikeutta asteikolla: erittäin helppo, helppo, normaali, vaikea ja erittäin vaikea. Taso-ontologiaa käytetään kuvaamaan kohdevideon vaikeutta (LOM-metadatan kenttä *lom/educational/difficulty*) ja asipitoisuuden laajuutta (LOM-metadatan kenttä *lom/educational/semanticDensity*).

3.5 Rooli

Tällä ontologialla ohjelmia luokitellaan sitä tavallisesti käyttävien henkilöiden mukaan (LOM-metadatan kenttä *lom/educational/intendedEndUserRole*). Mahdollisina arvoina

Opettaja ja Oppija.

Ideana on mahdollistaa tilanne, missä video on tarkoitettu esimerkiksi ala-astelaisten opettajille. Näin opettaja voi kouluttaa itseään tietyn kohderyhmän opettajana.

Tätä ontologiaa ei ole orava-projektissa hyödynnetty, koska katsottiin, että tästä ei ole hyötyä Klaffin metadatalalla.

3.6 Ohjelmatyyppi

Ohjelmatyyppi kuvaa ohjelman eli videon tyyppiä. Tämä viittaa videon esitystekniikkaan tai opetusmetodologiaan. Ohjelman tyyppi voi LOM:in mukaan olla aktiivinen, passiivinen tai yhdistelmä. Aktiivinen ohjelma edellyttää oppijalta aktiivista toimintaa, kun taas passiivinen ei tällaista toimintaa edellytä, ja yhdistelmä on näitten välimuoto. Tämä tieto löytyy LOM-tietomallin kentästä *lom/educational/interactivityType* (Klaffissa kenttä *lom/educational/learningResourceType*).

Näiden päätason käsitteiden alakäsitteet ovat tarkempia ja ne voivat olla esimerkiksi seuraavanlaisia: simulaatio, kysymys, harjoitustehtävä ja luento. Tämä tieto vastaa LOM-tietomallin kenttää *lom/educational/learningResourceType* (Klaffin kenttää *lom/educational/interactivityType*).

Tätä ontologiaa ei ole orava-projektissa hyödynnetty, koska asiakas katsoi sen olevan turha.

3.7 Ohjelmasarja

Ohjelmasarjat-ontologia koostuu ohjelmasarjoista. Ohjelmasarja on kokonaisuus, johon kuuluu ohjelmia.

Metadatatassa ohjelman kuulumista ohjelmasarjaan ilmaisee kentän *lom/relation/kind* arvo *isPartOf* ja ohjelmasarjan kenttä *lom/relation/target*. Tämän kentän ongelmana kuitenkin on sen epämääräisyys, koska se osoittaa välillä Opinportti-palvelun ohjelmasarjan kotisivulle, välillä ohjelmasarjan jakson sivulle ja toisinaan se on pelkkä sanallinen kuvaus. Tämän ikävä seuraus on, että *lom/relation/target* ei välttämättä ole sama samaan sarjaan kuuluville videoille, mikä vaikeuttaa kentän hyväksikäyttöä koneellisesti.

Tästä syystä ohjelmasarjaan kuulumisen ensisijainen tunnistusperuste on *lom/general/title*, joka on ohjelmasarjoihin kuuluvilla videoille tyypillisesti muotoa "ohjelmasarjan_nimi: videon nimi".

Ohjelmasarjat saadaan kentästä *lom/general/title* edellä osoitetulla tavalla, mutta näin saatuja tuloksia pyritään vielä hieman parantamaan hyödyntämällä kenttää *lom/relation/target*. Mikäli kentässä *lom/relation/target* esiintyy sana "jakso", voidaan kentän alkuosasta erottaa URL, jonka voidaan olettaa olevan sama kaikissa tähän sarjaan kuuluvissa videoissa. Näin voidaan yhdistää ohjelmasarjat, joiden kentän *lom/general/title* alkuosa eroaa, mutta jotka kentän *lom/relation/target* mukaan kuuluvat samaan sarjaan.

3.8 Ohjelma

Tämän ontologian tarkoituksena on identifioida kohdevideo. Ontologiassa kohdevideoita kutsutaan ohjelmaksi ja video on varsinainen URLilla osoitettava multimediaesitys. Ohjelmat-ontologia koostuu ohjelmista, jotka ovat uniikkeja.

Ohjelmaan liittyy monia ominaisuuksia, jotka kertovat asioita videosta. Tällaisia ovat esimerkiksi: shortdescription, longdescription, duration, difficulty, keyword, ageRange, intendedEndUserRole, context, partof, mediatype, programNumber, programSerie, semanticDensity, interactivityType, learningType.

Useimmat näistä ominaisuuksien arvoista saadaan melko suoraviivaisesti suoraan jostakin vastaavasta LOM-tietomallin mukaisesta kentästä. Toisissa taas suoritetaan hieman sovitusta, jotta ne saadaan sopimaan ontologioihin. Esimerkiksi difficulty saadaan kentästä *lom/educational/difficulty* ja sen arvo muunnetaan taso-ontologian mukaiseksi.

3.9 Teemalabels

Teemalabels on ontologia, joka määrittelee Klaffin avainsanat RDF-resursseiksi. Pääasiallinen tarkoitus on liittää näihin resursseihin nimi (label), joka näkyy käyttöliittymässä.

4 Metatiedot

Orava-portaalissa käytetään metatietoja paitsi Klaffin videoista, myös MuseoSuomen kohteista. Tässä luvussa listataan olennaiset käytetyt metatiedot.

4.1 Klaffin videot

Klaffin LOM-muotoisista metadatakorteista poimitaan RDF-muotoon käytettäväksi seuraavat tiedot.

- Avainsanat (<http://www.cs.helsinki.fi/group/orava/rdf#keyword>)
- Ohjelmasarjan nimi (<http://www.cs.helsinki.fi/group/orava/rdf#programSerie>)
- Ohjelman järjestysnumero ohjelmasarjassa (<http://www.cs.helsinki.fi/group/orava/rdf#programNumber>)
- Ohjelmatyyppi (<http://www.cs.helsinki.fi/group/orava/rdf#learningType>)
- Kohderyhmän ikä (<http://www.cs.helsinki.fi/group/orava/rdf#ageRange>)
- Käyttäjän rooli (<http://www.cs.helsinki.fi/group/orava/rdf#intendedEndUserRole>)
- Asiapitoisuus (<http://www.cs.helsinki.fi/group/orava/rdf#semanticDensity>)

- Vaikeusaste (<http://www.cs.helsinki.fi/group/orava/rdf#difficulty>)
- Oppimisympäristö (<http://www.cs.helsinki.fi/group/orava/rdf#context>)
- Mediatyyppi, video vai audio (<http://www.cs.helsinki.fi/group/orava/rdf#Mediatyyppi>)

4.2 MuseoSuomen kohteet

MuseoSuomen esinekorteista käytetään seuraavia tietoja.

- Asiasana (<http://www.cs.helsinki.fi/group/seco/ns/2004/03/18-esinekortti#asiasana>)
- Käyttöpaikka (<http://www.cs.helsinki.fi/group/seco/ns/2004/03/18-esinekortti#kaytopaikka>)
- Valmistuspaikka (<http://www.cs.helsinki.fi/group/seco/ns/2004/03/18-esinekortti#valmistuspaikka>)

5 Päätelysäännöt

Päätelysääntöihin kuuluvat näkymien kategoriapuut muodostavat kategoriasäännöt ja suosittusäännöt, jotka suosittelevat käyttäjälle kohteita videoiden yhteydessä. Päätelysäännöt löytyvät tiedostosta `ontodella/conf/fms/fms_sewehgrius_rules.pl` ja asetusparametreja säännöille löytyy tiedostosta `ontodella/conf/fms/parameters.pl`.

Päätelysääntöjen käytettävissä ovat ontologiat ja kohteet kuvaavat metadatakortit Orava- ja MuseoSuomi-projekteista.

Näkymiä toteutettiin yhdeksän kappaletta: *oppiaine*, *teema*, *ohjelmasarja*, *ohjelmatyyppi*, *kohderyhmä*, *rooli*, *vaativuus*, *oppimisympäristö* ja *mediatyyppi*. Näistä tosin *oppimisympäristö* ja *rooli* arvioitiin arkiston nykyisellä sisällöllä turhiksi ja jätettiin näyttämättä käyttöliittymässä. Lisäksi asiakas katsoi tarpeettomaksi *ohjelmatyyppi*-näkyvän.

Suosittelusääntöjä toteutettiin kuusi kappaletta. Näistä neljä suosittelee Klaffin videoita: *sama aihe*, *sarjan sama osa*, *sarjan seuraava* ja *sarjan edellinen*. Lisäksi kaksi sääntöä suosittelee kohteita MuseoSuomessa: *sama aihe* ja *sama paikka*.

Seuraavassa kuvataan kukin sääntö ja sen mahdollisesti käyttämät parametrit tarkemmin.

5.1 Kategoriasäännöt

Tässä luvussa kuvataan näkyvät toteuttavat kategoriasäännöt.

5.1.1 Oppiaine

Oppiaine-kategoria noudattaa ontologiassa määritettyä käsitehierarkiaa.

5.1.2 Teema

Teema-kategoria noudattaa ontologiassa määritettyä käsitehierarkiaa. Teema-kategoria täydentää oppiaine-kategoriaa tarjoamalla käyttäjän valittavaksi toisenlaisia avainsanoja. Näin vaikka molemmat näkymät käyttävätkin samoja metadatan avainsanoja, ne tarjoavat käyttäjälle mahdollisuuden tarkentaa hakuaan.

5.1.3 Ohjelmasarja

Ohjelmasarja-kategoria noudattaa ontologiassa määritettyä käsitehierarkiaa. Koska ohjelmasarjoja on paljon, ja näkymän kokoa käyttöliittymässä haluttiin rajoittaa, alikategorioiksi määritettiin sellaiset ontologian mukaiset kategoriat, joissa on vähintään tietty määrä lehtiä. Tätä vaadittua määrää lehtiä voi muuttaa asetuksella `parameter_n_leaves_required(N)`-parametrillä, missä N on sopiva kokonaisluku.

5.1.4 Ohjelmatyyppi

Ohjelmatyyppi-kategoria noudattaa ontologiassa määritettyä käsitehierarkiaa.

5.1.5 Kohderyhmä

Kohderyhmä-kategoria noudattaa ontologiassa määritettyä käsitehierarkiaa.

5.1.6 Rooli

Rooli-kategoria noudattaa ontologiassa määritettyä käsitehierarkiaa. Koska rooli-kenttää ei ole käytetty järkevästi videoiden metatiedoissa, rooli-näkymä jätettiin pois käyttöliittymästä.

5.1.7 Vaativuus

Vaativuus-kategoria muodostetaan vaikeus- ja asiapitoisuus-ontologioista. Vaativuus-kategoriaan on määritetty kolme alikategoriaa: *helppo*, *keskivaativa* ja *vaativa*.

Alikategorioiden lehdet määräytyvät vaikeus- ja asiapitoisuus-kentistä laskettavan pistemäärän perusteella. Pistelaskussa painotetaan vaikeus-tietoa, koska se kuvaa videon sisällän vaativuutta olennaisemmin. Vaikeus-arvo määrää $2/3$ ja asiapitoisuus $1/3$ vaativuus-pistemäärästä. Niille videoille, joille ei ole merkitty asiapitoisuutta, vaikeus määrää yksin vaativuuden.

Videon vaativuusalikategoriaan kuulumisen määrääviä pisterajoja voi säätää asetuksilla $\text{parameter_keskivaativa_points}(R_{kv})$ ja $\text{parameter_vaativa_points}(R_v)$, missä R_{kv} ja R_v ovat sopivia reaalilukuja väliltä $]0-5[$ siten, että $R_{kv} < R_v$.

5.1.8 Oppimisympäristö

Oppimisympäristö-kategoria noudattaa ontologiassa määritettyä käsittehierarkiaa. Koska oppimisympäristö-näkymää pidettiin turhana, se jätettiin pois käyttöliittymästä.

5.1.9 Mediatyyppi

Mediatyyppi-kategoria sisältää alakategoriat *ääni* ja *video*, joihin kohteet on jaettu sen perusteella, ovatko ne pelkkiä ääniklippejä vai videoita.

5.2 Suosittelemäärät

Kaikki suosittelemäärät näyttävät rajallisen määrän suosituksia videon yhteydessä. Kaikkiin videoihin ei välttämättä liity suosituksia. Suositusten määrää rajoitetaan sama aihe, sama aihe MuseoSuomessa ja sama paikka MuseoSuomessa -sääntöjen kohdalla parametrilla $\text{parameter_max_recommendations}(N)$ ja sarjan osien suosittelemääränsäntöjen kohdalla parametrilla $\text{parameter_max_recommendations_serie}(N)$, missä N on haluttu maksimimäärä suosituksia suositussääntöä kohti kokonaislukuna esitettynä.

5.2.1 Sama aihe

Sama aihe -sääntö suosittelee videon yhteydessä sellaisia muita videoita, joilla on videon kanssa eniten yhteisiä avainsanoja.

5.2.2 Sama osa sarjassa

Sama osa sarjassa -sääntö suosittelee videon yhteydessä sellaisia muita videoita, jotka kuuluvat videon kanssa saman sarjan samaan osaan.

Kuhunkin sarjan osaan kuuluu tyypillisesti useita videoita.

5.2.3 Seuraava osa sarjassa

Seuraava osa sarjassa -sääntö suosittelee videon yhteydessä sellaisia muita videoita, jotka kuuluvat videon kanssa saman sarjan seuraavaan osaan.

5.2.4 Edellinen osa sarjassa

Edellinen osa sarjassa -sääntö suosittelee videon yhteydessä sellaisia muita videoita, jotka kuuluvat videon kanssa saman sarjan edelliseen osaan.

5.2.5 Sama aihe MuseoSuomessa

Sama aihe MuseoSuomessa -sääntö suosittelee videon yhteydessä sellaisia kohteita MuseoSuomessa, joilla on eniten yhteisiä asiasanoja videon avainsanojen kanssa.

5.2.6 Sama paikka MuseoSuomessa

Sama aihe MuseoSuomessa -sääntö suosittelee videon yhteydessä sellaisia kohteita MuseoSuomessa, joiden käyttö- tai valmistuspaikasta löytyy eniten vastaavuuksia videon avainsanojen kanssa.

6 Käyttöliittymä

Orava-portaalin käyttöliittymän rakenne ja toiminnallisuus toteutettiin XSLT-muunnoksien. XSLT-tiedostot käsittelevät XML-tiedostoja ja tekevät niiden pohjalta määrätynlaisen HTML-sivun. Lisäksi käyttöliittymän värien, kirjasinten ja muiden puhtaasti ulkoasullisten seikkojen toteuttamiseksi käytettiin CSS-tyylitiedostoa.

6.1 XSLT-muunnokset

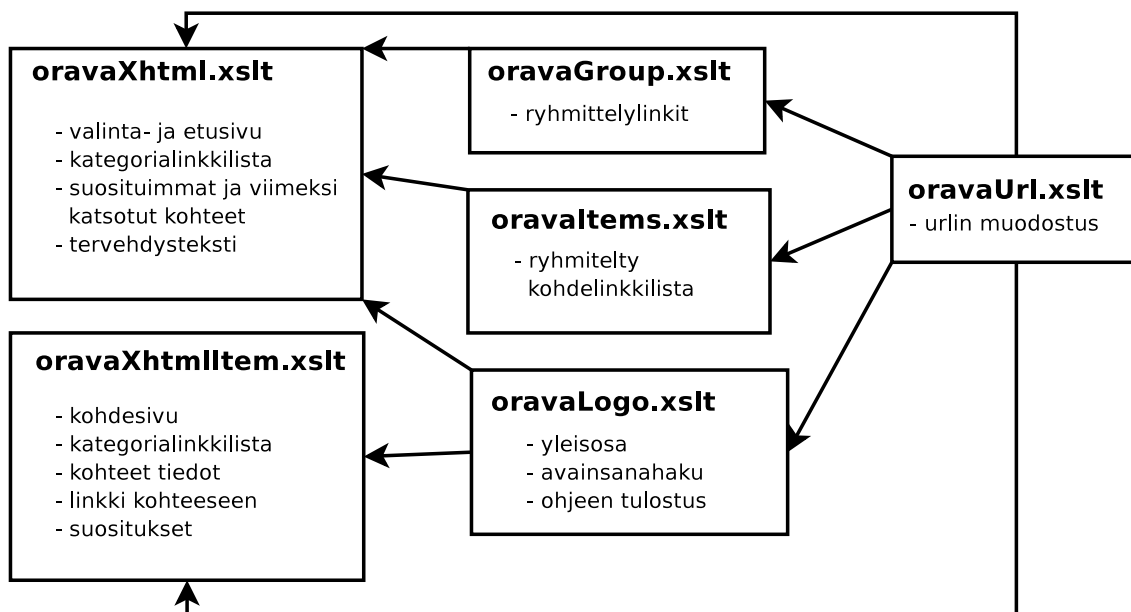
Orava-portaaliin kuuluu kaikkiaan kuusi XSLT-tiedostoa, joita käytetään nimenomaan käyttöliittymän muodostamiseen. Kaikki käyttöliittymään kuuluvat XSLT-tiedostot sijaitsevat hakemistossa `ontoviews/webapp/transformation`. Kuvassa 1 on kuvattu XSLT-tiedostojen väliset suhteet ja niiden tärkeimmät tehtävät.

6.1.1 oravaXhtml.xslt

oravaXhtml.xslt jakautuu toiminnallisuudeltaan kolmeen osaan.

Ensinnäkin oravaXhtml.xslt kokoaa yhteen koko käyttöliittymän toiminnallisuuden silloin, kun käyttäjä on etu- tai valintasivulla. Tiedostosta löytyy HTML-sivun runko ja taulukko, jonka sisälle käyttöliittymä muodostetaan. oravaXhtml.xslt käyttää apunaan tiedostoja oravaGroup.xslt, oravaLogo.xslt ja oravaItem.xslt ryhmittelylinkkien, yleisosan ja kohdelinkkien tulostamisessa.

Toisekseen oravaXhtml.xslt sisältää templaattit etu- ja valintasivulla olevan kategorialinkkilistan muodostamiseen rekursiivisesti. Linkkilistaan kuuluu varsinaisten kategorioiden



Kuva 1: XSLT-tiedostojen väliset suhteet.

valinta- ja poistolinkkien ohella myös avainsanojen poistolinkit silloin, kun käyttäjä on tehnyt avainsanahaun tai useita.

Kolmanneksi oravaXhtml.xslt-tiedostosta löytyy templaattit, joilla tulostetaan etusivulle tervehdysteksti sekä kohdelinkit suosituimpiin ja viimeksi katsottuihin kohteisiin.

oravaXhtml.xslt käyttää valinta-, poisto- ja kohdelinkkien href-parametrin muodostamisessa oravaUrl.xslt-tiedostossa sijaitsevia templaatteja.

Jos käyttäjä on klikannut ohje-linkkiä ollessaan etu- tai valintasivulla, oravaXhtml.xslt kutsuu oravaLogo.xslt-tiedostossa olevaa käyttöohjeen tulostus -templaattia. Vastaavasti jos ohje ei ole päällä, oravaXhtml.xslt tulostaa sen tilalle normaalin etu- tai valintasivun.

6.1.2 oravaXhtmlItem.xslt

Myös oravaXhtmlItem.xslt jakautuu toiminnallisuudeltaan kolmeen osaan.

Ensinnäkin oravaXhtmlItem.xslt kokoaa yhteen koko käyttöliittymän toiminnallisuuden silloin, kun käyttäjä on kohdesivulla. Tiedostosta löytyy HTML-sivun runko ja taulukko, jonka sisälle käyttöliittymä muodostetaan. oravaXhtmlItem.xslt käyttää apunaan tiedostoa oravaLogo.xslt käyttöliittymän yleisosan tulostamiseen.

Lisäksi oravaXhtmlItem.xslt sisältää templaattit kohdesivulla olevan kategorialinkkilistan muodostamiseen rekursiivisesti. Kategorialinkkien href-parametrin muodostamisessa käytetään oravaUrl.xslt-tiedostossa sijaitsevia templaatteja.

Kolmanneksi oravaXhtmlItem.xslt-tiedostosta löytyy templaattit, joilla tulostetaan varsinainen kohdesivu: videon nimi, esittelyteksti, linkki videoon tai äänileikkeeseen sekä lin-

kit suositeltuihin kohteisiin sekä Orava-portaalissa että MuseoSuomessa.

Jos käyttäjä on klikannut ohje-linkkiä ollessaan kohdesivulla, oravaXhtmlItem.xslt kutsuu oravaLogo.xslt-tiedostossa olevaa käyttöohjeen tulostus -templaattia. Vastaavasti jos ohje ei ole päällä, oravaXhtmlItem.xslt tulostaa sen tilalle normaalin kohdesivun.

6.1.3 oravaItems.xslt

oravaItems.xslt on aputiedosto, joka sisältää templaattit, joiden avulla voidaan muodostaa kohdelinkkilista valintasivulle. Se tulostaa käyttöliittymään tekstin, jossa kerrotaan, minkä kategorian mukaan linkit on ryhmitelty. Lisäksi se tuottaa ryhmitellyn listan kohteista: kategorioiden nimet, niihin kuuluvat kohteet sekä kohteiden selailuun tarkoitettut edellinen- ja seuraava-linkit, mikäli yhteen kategoriaan kuuluu enemmän kuin kaksikymmentä kohdetta. Sekä edellinen- ja seuraava-linkkien että varsinaisten kohdelinkkien href-parametrien tekoon oravaItems.xslt käyttää apunaan oravaUrl.xslt-tiedoston templaatteja.

6.1.4 oravaGroup.xslt

oravaGroup.xslt on aputiedosto, jonka sisältämien templaattien avulla voidaan muodostaa ryhmittelylinkit. Templaatti tulostaa tekstin, jossa kehoitetaan ryhmittelemään kohteet uudelleen, sekä tulostaa tekstin alle ryhmittelylinkit. Ryhmittelylinkkien href-parametrien muodostamisessa käytetään oravaUrl.xslt-tiedoston templaatteja.

6.1.5 oravaLogo.xslt

oravaLogo.xslt on aputiedosto, jonka templaattit tulostavat suunnitteludokumentissa määritellyn yleisosan. Yleisosa sisältää linkit uuteen hakuun, ohjeeseen ja palautteeseen. Lisäksi siihen kuuluu avainsanahaun lomake ja järjestelmän logo. oravaLogo.xslt:ssä on joitakin templaatteja, joiden avulla avainsanahaun lomakkeeseen saadaan muodostettua oikeanlaiset input-kentät. Lisäksi tiedostossa on templaatteja, joiden avulla käyttöohje saadaan näytettäväksi tai pois näkyvistä.

Yleisosan lisäksi oravaLogo.xslt sisältää templaatin, jonka avulla voidaan tulostaa käyttöohje.

Uusi haku -linkin href-parametri muodotetaan oravaUrl.xslt-tiedoston templaattien avulla.

6.1.6 oravaUrl.xslt

oravaUrl.xslt-aputiedoston templaattien avulla voidaan muodostaa käyttöliittymän linkkien href-parametreihin oikeanlaiset urlit.

6.2 CSS-tyylit

Orava-portaaliin kuuluu CSS-tiedosto - orava.css - jossa on määritelty käyttöliittymän värit, kirjasinten koot ja tyylit sekä muut puhtaasti ulkoasulliset ominaisuudet. Muokkaamalla orava.css-tiedostoa voidaan muokata myös käyttöliittymän ulkoasua koskematta varsinaisiin XHTML-tiedostoihin. orava.css ei vaikuta käyttöliittymän kokonaisrakenteeseen.

orava.css-tiedosto löytyy hakemistosta `ontoviews/webapp/content/stylesheets/` ja käyttöliittymään sijoitetut kuvat hakemistosta `ontoviews/webapp/content/images/`.

6.3 Käyttöliittymän toteutus

Orava-portaalin käyttöliittymä eroaa valmiina hieman suunnitteludokumentissa määritellystä käyttöliittymästä. Kuvasta selviää, miten suunnitteludokumentissa määritellyt käyttöliittymän osat sijoitettiin Oravaan. Lisäksi tässä kuvataan, miltä osin niiden toteutus eroaa suunnitelmasta.

6.3.1 Yleisosa

Yleisosa toteutettiin kuten suunnitteludokumentissa oli määritelty sillä erolla, että järjestelmän logo ei toimi linkkinä uuteen hakuun.

6.3.2 Etusivu

Järjestelmän etusivu ja siinä oleva elävä linkkilista toteutettiin kuten suunnitteludokumentissa oli määritelty. Suosituimmat kohteet ja viimeksi katsotut kohteet -linkkilistoihin ei ole laitettu kuvia, vaikka tämä mahdollisuus mainittiin suunnitteludokumentissa. Lisäksi järjestelmän etusivulla on tervehdysteksti, jota ei mainittu suunnitteludokumentissa. Kuvasta 2 näkyy, miten suunnitteludokumentissa mainitut käyttöliittymän osat on sijoiteltu Oravan etusivulle.

6.3.3 Valintasivu

Valintasivu ja siihen liittyvä elävä linkkilista on toteutettu pääpiirteissään juuri niin kuin suunnitteludokumentissa määriteltiin. Kuvasta 3 näkyy, miten suunnitteludokumentissa mainitut käyttöliittymän osat on sijoiteltu valintasivulle.

Kohdelinkkilistan toteutuksessa on joitakin eroja suunnitelmaan verrattuna. Kuten suunnitteludokumentissa sanottiin, kohteet ryhmitellään valitun kategorian alakategorioiden mukaan. Kuitenkin niille kohteille, jotka eivät kuulu mihinkään alakategoriaan, tehdään sivun alkuun oma ryhmä, jolla on nimenä valitun kategorian nimi. Lisäksi, jos valitulla kategoriolla ei ole alakategorioita, kohteet ryhmitellään valitun kategorian mukaan. Kohdelinkkilista eroaa suunnitelmasta myös siten, että kategorian nimen perässä ei ole tekstiä,



Kuva 2: Oravan etusivun rakenne.

jossa ilmoitettaisiin, kuinka monta kohdetta ryhmään kuuluu ja kuinka monta kohdelinkkiä on näkyvissä (esimerkiksi 20/754).

Elävässä linkkilistassa ei näytetä kategoriaan kuuluvien kohteiden lukumäärää aina. Se näytetään, jos kategoria sisältää haettuja kohteita ja jos kategoria ei ole valitun kategorian (esi)vanhempi.

Ryhmittelylinkkejä ei lopullisessa käyttöliittymässä sijoitettu valintasivun yläreunaan vaan sen oikeaan reunaan. Lisäksi ryhmittelylinkkeinä eivät toimi pelkästään valitut kategoriat vaan myös ne valitsemattomat kategoriat, joihin valittuihin kategorioihin kuuluvat kohteet kuuluvat. Sitä kategoriaa, jonka mukaan kohteet on ryhmitelty, ei esitetä lainkaan - ei edes tekstinä - muiden ryhmittelylinkkien rinnalla. Sen sijaan valintasivun yläreunassa on teksti, joka kertoo, minkä kategorian mukaan linkit on ryhmitelty.

Lisäksi valintasivun kohdelinkkien perässä on enintään 200 merkkiä pitkä kohdetta kuvaileva teksti, jota ei mainittu suunnitteludokumentissa. Yli 200 merkkiä sisältävät esitelytekstit on katkaistu ja niiden perään on laitettu kolme pistettä.

6.3.4 Kohdesivu

Kohdesivu ja siihen liittyvä elävä linkkilista on toteutettu pitkälle suunnitteludokumentin mukaisesti. Kohdesivulla ei kuitenkaan mainita kohteen sijaintia. Käyttäjälle ei myöskään näytetä kyseistä kohdetta, vaan tarjotaan pelkästään linkki, jota painamalla kohteen voi avata erilliseen mediasoitto-ohjelman ikkunaan. Linkkitekstistä selviää, onko kohde video vai äänileike. Kuvasta 4 näkyy, miten suunnitteludokumentissa mainitut käyttöliittymän osat on sijoiteltu valintasivulle.

Kohteesta kerrottavat tiedot rajoittuvat valmiissa käyttöliittymässä kohteen nimeen ja esitelytekstiin. Lisäksi kohdesivulla on linkki, josta voi siirtyä kohteeseen liittyvän Ylen oppimateriaalin pariin, mikäli sellainen on olemassa.

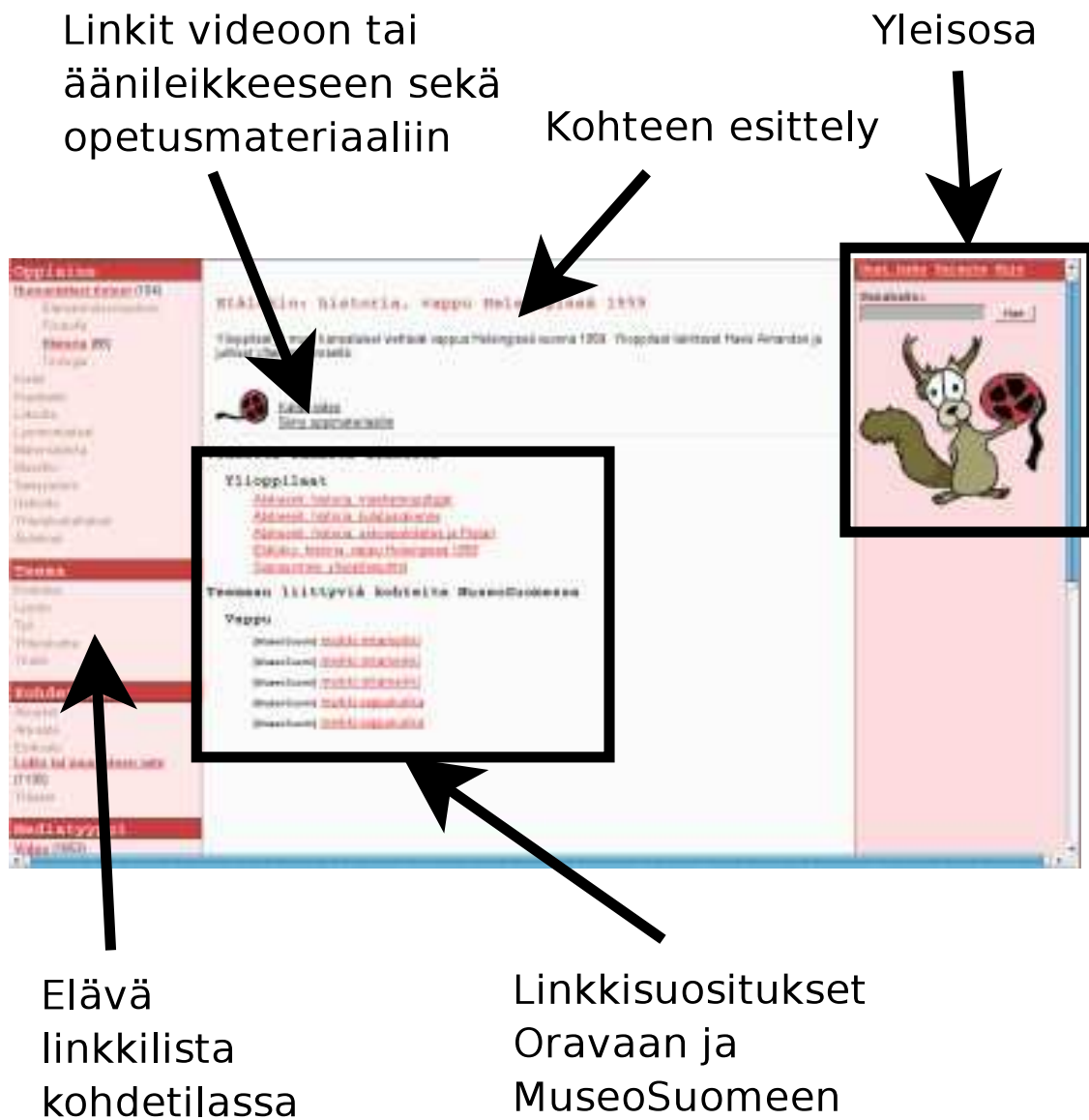
Kohdesivulle ei toteutettu linkkejä seuraavaan ja edeltävään käyttäjän valintaan kuuluvaan kohteeseen. Katsottiin, että samaan sarjaan kuuluvien seuraavan ja edellisen videon suosittelulinkit korvaavat ne.

Suosittelulinkit toteutettiin suunnitteludokumentin mukaisesti. Linkkejä ei ole kuitenkaan jaoteltu suositteluperusteen alle Oravan kategorioiden mukaan vaan esimerkiksi aiheiden tai osanumeroiden mukaisesti. Myös MuseoSuomesta suositellut kohteet on jaoteltu vastaavalla tavalla.

6.3.5 Avainsanahaku

Portaalissa toteutettiin myös avainsanahaku, jonka toiminnallisuutta ei kuvattu suunnitteludokumentissa. Tehtyään avainsanahaun yleisosan lomakkeella, käyttäjä näkee käyttämänsä avainsanahaut linkkeinä käyttöliittymän vasemmassa yläkulmassa kategorialinkkien yläpuolella. Painamalla linkkiä käyttäjä voi poistaa avainsanan.

Kun käyttäjä tekee avainsanahaun, kohteet ryhmitellään Mediatyyppi-kategorian mukaan.



Kuva 4: Oravan kohdesivun rakenne.

6.3.6 Ohje

Orava-portaalissa ohje toteutettiin siten, että se upotetaan sivulle. Käyttäjä saa ohjeen esiin painamalla yleisosan ohje-linkkiä ja vastaavasti poistettua painamalla ohje pois -linkkiä. Ohje pois -linkki tulee ohje-linkin tilalle, jos käyttäjä katsoo ohjetta. Muuten käyttöliittymässä näkyy vain ohje-linkki.

Kun käyttäjä on etusivulla, ohje tulostetaan tervehdystekstin ja viimeksi katsottujen ja suosituimpien kohteiden kohdelinkkien tilalle käyttöliittymän keskiosaan. Kun käyttäjä on valintasivulla, ohje korvaa kohdelinkkilistan sivun keskiosassa. Kohdesivulla ohje tulee keskelle kohteen tietojen ja suosittelulinkkien tilalle.

Kun käyttäjä avaa ohjeen, sivun yleisosa, kategorialinkkilista ja mahdolliset ryhmittelylinkit pysyvät ennallaan. Käyttäjä voi edelleen tehdä tai poistaa valintoja tai ryhmitellä kohteita painamalla linkkejä. Tällöin ohje katoaa näkyvistä ja kulloinenkin sivu näytetään käyttäjälle kuten normaalistikin.

7 Ontoviews muutokset

Orava-portaali rakennettiin Ontoviews-järjestelmän päälle, joten pääpiirteittäin se toimii Ontoviewsien mukaisesti. Joitakin muutoksia kuitenkin tehtiin.

7.1 Ustategen.xsp

Ontoviewsien Cocoon liukuhihnojen (pipeline) juurena toimii ustateGen.xsp-ohjelma. Tämä ohjelma tekee xml-dokumentin, joka sisältää käyttäjän valinnat, asetukset ja muuta tietoa portaalista. Orava projektissa siihen lisättiin ohjesivua varten help-parametrin käsittely, ja queryString-elementin tulostaminen tuotettavaan xml-dokumenttiin. Lisäksi lisättiin kirjanpito suosituimmista kohteista ja näiden tulostus tuotettavaan xml-dokumenttiin. Suosituimpia kohteita ei tallenneta mihinkään jos palvelin sammutetaan.

Ohjelmassa muutettiin myös oletusarvoista ryhmittelykategoriaa. Haluttiin välttää avainsanan mukaan ryhmittelyä, koska tällöin seuraava- ja edellinen-linkit eivät toimineet ja lisäksi avainsanan mukaan ryhmittely ei tarjonnut käyttäjälle mitään hyödyllistä informaatiota. Jos käyttäjä ei ole valinnut ryhmittelykategoriaa niin käyteetään ensin käyttäjän valitsemia muita kategorioita ja jos näitä ei ole, eli on tehty pelkkä avainsanahaku, ryhmitellään mediatyyppi-näkymän mukaan. Näkymäksi valittiin juuri mediatyyppi, koska se on säännöllisesti jokaisella kohteella.

Lisäksi ustategen.xsp:stä poistettiin ominaisuus, joka mahdollisti sähköpostin lähettämisen antamalla sopivia parametreja http-pyyntöön. Tämä tehtiin, koska kyseistä ominaisuutta ei tarvittu mihinkään ja se oli melko vakava turvallisuusongelma.

7.2 OrvTransform

Yhtenä muuntimena Ontoviewsin Cocoon liukuhihnoissa on OrvTransform, joka on Javalla toteutettu muuntaja. Se muuntaa Ontogator-kyselyn tuloksen muotoon, jonka jatkokäsittelyminen xsl-tyylitiedostoissa on helpompaa. Näin itse käyttöliittymän tekevistä xsl-tyylitiedostoista saatiin yksinkertaisempia. Tämä muunnos päätettiin tehdä Javalla xsl:n sijaan sen takia, että xsl ei sovellu hyvin kovin monimutkaisten operaatioiden suorittamiseen. Vastaavan muunnoksen tekeminen xsl:llä olisi ollut huomattavasti työläämpi tehdä ja todennäköisesti suorituskyvyltäänkin se olisi ollut heikompi.

8 Asennusskriptit

Asennusskriptit on toteutettu käyttäen POSIX-standardin mukaista komentotulkkia (shell).

Skriptit `./start-ontodella` ja `./start-ontoviews` ovat suoraviivaisia skriptejä, joilla käynnistetään vastaavat ohjelmiston osat. Ontodellan käynnistykseen liittyy konfiguraatitiedosto `.ontodellarc`, jonka `./build` generoi automaattisesti funktiossa `ontodella_setup`.

Järjestelmään olennaisin skripti on `./build`, jolla hoidetaan järjestelmän kääntäminen, käyttöönotto ja erinäisiä päivitystoimenpiteitä. Skriptiä `./build` tarkastellaan seuraavaksi.

Mikäli ohjelman jatkokehityksessä tai muusta ennalta arvaamattomasta syystä tarvitaan uusia java-kirjastoja, voidaan ne helposti ottaa käyttöön lisäämällä jar-paketit hakemistoon `lib-java/`. Näin build-skripti ottaa nämä jar-paketit automaattisesti huomioon Java-ohjelmia käännettäessä ja ajettaessa. Tämä tapahtuu funktiossa `set_java`. Ontoviews tarvitsee monia sille spesifisiä java-kirjastoja, minkä vuoksi Ontoviewsin java-kirjastot asetetaan omassa funktiossa `set_javax`.

Jotta konsolin tekstitulva voitaisiin pitää hallinnassa ohjataan ohjelmien tulosteet muuttujan `NORMALREDIRECT` (`stdout`, normaalin tekstin ohjaus) avulla. Tämä parametri asetetaan skriptin parametrien mukaisesti, jolloin on mahdollista kontrolloida skriptin tulostusta.

Jokaiselle toimenpiteelle, kuten komponentin kääntämiselle tai ajamiselle, on oma funktio, joita yhdistelemällä on luotu loogisia kokonaisuuksia. Esimerkiksi `./build prepare_data`, joka ajetaan uusien LOM-muotoisten Klaffi-tiedostojen lisäyksen jälkeen koostuu seuraavista vaiheista: asetetaan sopiva `CLASSPATH` javalle, korjataan virheelliset Klaffi-tiedostot, ajetaan `xml2rdf` tuottaen RDF:ää, alustetaan ontodella luomalla sille konfiguraatitiedosto ja kopioimalla ontologiat ontodellan käyttämään polkuun ja lopuksi ontodella käyttäen generoidaan OntoViewsin käyttämä lopullinen metadata.

9 Vaatimukset

Tässä luvussa esitetään lista vaatimusedokumentin [VD] vaatimuksista ja havaitaan, että toteutettu järjestelmä toteuttaa ne kaikki.

- T-1 Luvussa 6 on kuvattu, että järjestelmällä on etusivu, jossa näkyvät ylimmän tason kategoriat.
- T-2 Luvussa 6 on kuvattu, että järjestelmä antaa käyttäjälle mahdollisuuden rajoittaa näytettyä moninäköä selailemalla tai avainsanahauulla.
- T-3 Luvussa 6 on kuvattu, että järjestelmä näyttää käyttäjälle tämän tekemät valinnat ja niiden yhteydessä linkin, jota seuraamalla kyseinen valinta poistetaan. Lisäksi järjestelmä näyttää linkin, jota seuraamalla kaikki valinnat poistetaan.
- T-4 Luvussa 6 on kuvattu, että käyttäjä voi suorittaa haun avainsanalla.
- T-5 Luvussa 6 on kuvattu, että käyttäjä voi lisätä kategorioita valintoihinsa seuraamalla moninäkömön linkkejä.
- T-6 Luvussa 6 on kuvattu, että käyttäjä voi valita kohteen, jolloin järjestelmä näyttää hänelle kohdesivun, josta löytyy videon kuvaus sekä linkki siihen.
- T-7 Luvussa 6 on kuvattu, että järjestelmä näyttää käyttäjälle näkymiä, joihin liittyy kategorioita, joihin liittyy alakategorioita ja/tai kohteita. Näytetyt kohteet liittyvät kategorioihin, joita käyttäjä ei ole rajannut pois valinnoillaan.
Luvuissa 3 ja 5 on kuvattu, että näkymät, kohteet ja kategoriat ovat linkitetty keskenään ontologioiden ja päättelysääntöjen avulla.
- T-8 Luvussa 6 on kuvattu, että järjestelmä suosittelee käyttäjälle muita hänen valitsemaan kohteeseen liittyviä kohteita. Luvussa 5 on kuvattu, että nämä suositukset tehdään päättelysääntöjen avulla.
- T-9 Luvussa 8 on kuvattu, että ylläpitäjä voi generoida Ontoviews-ohjelmiston ymmärtämää RDF-muotoista metadattaa LOM-muotoisista metadatakorteista (metadatatiedoista).
- T-10 Luvussa 6 on kuvattu, että käyttäjä voi lähettää ylläpitäjälle palautetta.
- T-11 Luvuissa 6 ja 5 on kuvattu, että käyttäjän valittua kohteen järjestelmä suosittelee hänelle valittuun kohteeseen liittyviä kohteita MuseoSuomi-portaalista.
- L-1 Luvussa 3 on kuvattu, että ontologioissa on käytetty opetukseen liittyviä tietoja videoiden metadatasta.
- L-2 Luvussa 2 on kuvattu, että XML2RDF-työkalu on toteutettu helposti muunnettavaksi ja ylläpidettäväksi.

- Y-1 Luvussa 5 on kuvattu, että järjestelmän käytettävissä on MuseoSuomi-portaalin metadataa, jolla järjestelmä voi suositella käyttäjälle kohteita MuseoSuomi-portaalista.
- Y-2 Luvussa 2 on kuvattu, että Klaffi-materiaalin metadata muunnetaan RDF/XML-muotoon.
- Y-3 Orava-portaali toteutettiin niin, että se toimii Helsingin Yliopiston tietojenkäsittelytieteen laitoksen linux-ympäristössä.

10 Tunnetut ohjelmavirheet

Ohjelma ei sisällä tunnettuja virheitä.

Lähteet

- Klaffi Yleisradio oy:n klaffi-opetusvideoarkisto.
<http://www.yle.fi/klaffi>. [14.2. 2005]
- LOM Learning object metadata. <http://ltsc.ieee.org/wg12/>.
[14.2. 2005]
- OntoViews Ontoviews. <http://www.cs.helsinki.fi/group/seco/museums/dist/>.
[14.2. 2005]
- SD Orava, Suunnitteludokumentti.
- VD Orava, Vaatimusdokumentti.
- YD Orava, Ylläpitodokumentti.
- SemWeb Semantic web. <http://www.w3.org/2001/sw/>. [19.2. 2005]
- URI Uniform resource identifier. <http://www.ietf.org/rfc/rfc3986.txt>.
[10.3. 2005]
- XML Extensible markup language. <http://www.w3c.org/XML/>.
[14.2. 2005]
- XSL Extensible stylesheet language. <http://www.w3c.org/XSL>.
[20.2. 2005]
- MuseoSuomi MuseoSuomi. <http://museosuomi.cs.helsinki.fi/>.
[14.2. 2005]
- RDFS Rdf schema. <http://www.w3.org/TR/rdf-schema/>. [10.3. 2005]

- RDF Resource description framework. <http://www.w3c.org/RDF/>.
[14.2. 2005]
- Opinportti Yle opinportti. <http://www.yle.fi/opinportti>. [21.2. 2005]