Projektisuunnitelma

WWW-museo

Helsinki 13. helmikuuta 2002

Versio 1.0

HELSINGIN YLIOPISTO
Tietojenkäsittelytieteen laitos

Sisältö

1 Johdanto

Projektissa on tarkoitus tuottaa selainpohjainen dynaaminen WWW-sivusto, eli virtuualimuseon näyttely. Näyttelyssä vierailija voi käydä läpi tietokannassa olevia kuvia. Näyttelyn on tarkoitus olla älykäs, eli ohjelman pitäisi osata päätellä vierailijan kiinnostuksen kohteet ja ehdottaa sopivia kuvia. Näyttelyn teemana on tohtoripromootio.

1.1 Asiakas

Projektin asiakas on Eero Hyvönen.

2 Projektin tavoitteet

2.1 Ohjelmisto

Tavoitteena on tuottaa toimiva ohjelmisto, joka toetuttaa vähintään asiakkaan haluamat perustoiminnot. Ohjelmiston tulisi olla vakaa, eli se ei saa kaatua. Käyttöliittymästä pyritään tekemään mahdollisimman selkeä, jotta näyttelyssä vierailu olisi helppoa.

2.2 Lisäominaisuudet

Oletusarvoisesti asiakaan ehdottamia lisäominaisuuksia ei pyritä totetuttamaan. Projektissa käytettävät ohjelmat ja teknologiat ovat projektiryhmälle uusia, joten niihin tutustuminen vie aikaa. Jo perustoimintojen toteuttamisessa on paljon työtä.

2.3 Rajoittavia tekijöitä

Yliopiston mseon tietokannasta saaduissa kuvissa olevan metatiedon määrä vaihtelee paljon kuvakohtaisesti. Tekoälyn taso riippuu metatiedon määrästä. Jos kuvista ei ole annettu mitään taustatietoja, tekoälyalgoritmin on vaikea päätellä kuvista ja niiden yhteyksistä yhtään mitään. Taustatiedon määrä on lähes suoraan verrannollinen järjestelmän toimivuuteen ja älykkyyteen.

3 Toteutus- ja toimintaympäristö

3.1 Laitteisto ja tarvittavat ohjelmat

Ohjelmiston tulisi toimia laitteistoriippumattomasti verkossa selaimen välityksellä. Käyttöliittymässä käytetään mahdollisesti JavaScriptiä ja Java-sovelmia. Ohjelmistoa käyttävässä järjestelmässä tulisi olla asennettu Java-tuki. Käyttöliittymä on graafinen.

3.2 Käyttäjiltä vaadittavat taidot

Ohjelmiston käyttäminen ei vaadi erityistaitoja. Selaimen ja graafisen käyttöliityymän toimintojen ymmärtäminen riittää.

4 Projektin henkilöt ja tehtäväjako

4.1 Projektiryhmä

Samppa Saarela samppa.saarela@cs.helsinki.fi
Kim Josefsson kim.josefsson@helsinki.fi
Eetu Mäkelä eetu.makela@cs.helsinki.fi
Robert Holmberg robert.holmberg@helsinki.fi
Pasi Lehtimäki pjlehtim@cs.helsinki.fi
Taneli Rantala Taneli.Rantala@cs.helsinki.fi

Projektiryhmä on johtajavetoinen ja projektipäällikkö on Samppa Saarela. Muutkin ryhmän jäsenet voivat tarvittaessa johtaa ryhmää, jos heillä on erityisosaamista tietyltä ohjelmiston osa-alueelta.

4.2 Ohjaaja ja asiakas

Ryhmän ohjaaja on Matti Nykänen ja asiakas Eero Hyvönen.
Yhteystiedot:

Matti Nykänen

E-mail: Matti.Nykanen@cs.helsinki.fi
Huone: C481

Eero Hyvönen
E-mail: Eero.Hyvonen@cs.helsinki.fi
Huone: A405

4.3 Tehtäväjako

Eri tehtäviä pyrittämään hajauttamaan mahdollisimman paljon siten, että jokainen osallistuu kaikkiin työvaiheisiin. Projektipäällikkö päättää tehtävien jakamisesta. Ryhmän jäsenet kirjaavat ylös työtuntinsa, jotta projektipäällikkö pystyy seuraamaan projektin edistymistä.

Jokainen dokumentoi oman koodinsa. Dokumenteilla ja jokaisella ohjelmiston osa-alueella on oma vastuuhenkilönsä. Alustavat vastuuhenkilöt eri alueille ovat:

Dokumentit ja XML-asiat	Kim
Käyttöliittymä	Pasi ja Samppa
Tietokanta-asiat	Robert
Tekoäly	Eetu ja Taneli

Sihteeri valitaan kokouksissa viikoksi kerrallaan. Sihteeri pitää kokouspöytäkirjaa ja laittaa sen verkkoon.

5 Projektin toiminta

5.1 Projektimalli

Projektimalliksi valittiin lineaarinen vesiputousmalli. Seuraavassa on eritelty eri projektivaiheiden arvioidut kestoajat:

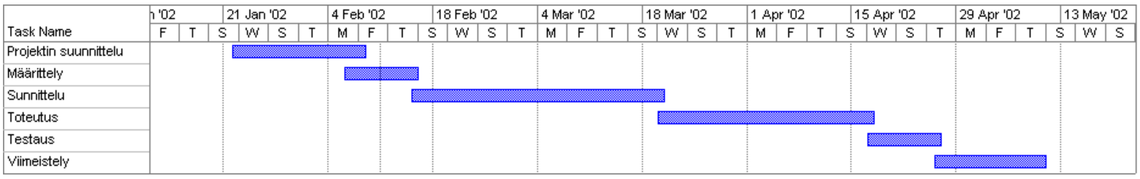
1)	Projektin suunnittelu	2.5vk
2)	Määrittely	1vk
3)	Suunnittelu	4.5vk
4)	Toteutus	4vk
5)	Testaus	1.5vk
6)	Viimeistely	2vk

5.2 Tarkistukset

Jokaisen vaiheen dokumentit tarkastetaan ja hyväksytään ennen siirtymistä seuraavaan vaiheeseen. Määrittelydokumentin ja suunnitteludokumentin valmistuttua voidaan pitää kokous asiakkaan kanssa. Projektin etenemistä ja aikataulussa pysymistä tarkkaillaan kokouksissa. Myös ryhmän jäsenten työmäärää arvioidaan säännöllisesti.

5.3 Aikataulu

Seuraavalla sivulla esitetään projektin suunniteltu aikataulu ensin Gantt-kaaviona ja sitten taulukkona:



Kuva 1: Projektin aikataulu

Projektin suunnittelu	22.1 – 8.2
Määrittely	8.2 – 15.2
Suunnittelu	15.2 – 20.3
Toteutus	20.3 – 17.4
Testaus	17.4 – 26.4
Viimeistely	26.4 – 10.5

Aikataulussa jää pelivaraa lähinnä suunnitteluosan loppuun sekä viimeistelyosuuteen. Toteutusosuudessa täytyy huomioida pääsiäisloma ja sen vaikutus. Ohjelmisto on valmis viimeistään 10.5. Viimeistelyvaihe käytetään koodin hiomiseen ja loppuraportin kirjoittamiseen. Aikaa voidaan käyttää myös muihin työvaiheisiin, mikäli aikataulusta ollaan jäljessä. Valmis ohjelma esitellään asiakkaalle erikseen sovittavana ajankohtana.

6 Projektin välineet ja menetelmät

6.1 Dokumentointi

Dokumentointiin käytetään LaTeX:ia ja dokumentit tallennetaan ps-muodossa. Dokumenttien ulkoasu pidetään yhtenevänä koko projektin ajan.

6.2 Versionhallinta

Versionhallintaan käytetään laitoksen Linux–järjestelmässä toimivaa CVS–versionhallintajärjestelmää. CVS:n alla pyritään pitämään kaikki ryhmän tuottama materiaali.

6.3 Suunnittelu

Suunnittelussa käytetään Ohjelmistotuotanto-kurssin kaaviotekniikoita. Niiden piirtämisessä voidaan käyttää apuna laitokselta löytyviä ohjelmia.

6.4 Ohjelmointi

Ohjelmoinnissa käytetään Java-kielen versiota 1.3. Ohjelmistossa käytetään Java Server Pages –järjestelmää, jota varten ryhmän käyttöön asennetaan Tomcat. XML-tietokantana käytetään DBXML:ää. Xercesiä käytetään XML–parserina. Järjestelmää nimeltä Jena saatetaan käyttää RDF–rakenteen toteuttamiseen. Käyttöliittymä ohjelmoidaan HTML:llä, johon liitetään mahdollisesti JavaScriptiä tai Java–sovelmia. Koodi koodataan englanniksi, mutta kommentoidaan suomeksi. Koodi kommentoidaan siten, että koodista voidaan automaattisesti generoida Javadoc–sivusto.

7 Riskienhallinta

Riski: Ryhmän jäsen ei tee tarpeeksi töitä
Miten havaitaan: Seurantakokuksien ja tuntikirjanpidon perusteella
Miten ratkaistaan: Järjestellään tehtävät uudestaan. Mikäli tämä ei auta, projektipäällikkö pitää jäsenelle puhuttelun ja motivoi tätä.

Riski: Työvaiheet työläämpiä kuin on osattu ennakoida
Miten havaitaan: Suunnitellun aikataulun ja projektin etenemisen vertaileminen
Miten ratkaistaan: Karsitaan ominaisuuksia tai kiristetään tahtia. Neuvottelu asiakkaan kanssa

Riski: Tuntemattomien tekniikoiden (mm. RDF) opiskelu liian hidasta
Miten havaitaan: Kesustellaan kokouksissa
Miten ratkaistaan: Ryhmässä olevat mahdolliset asiantuntijat opastavat muita. Hyvien ohjeiden ja oppaiden etsiminen.

Riski: Laitoksen järjestelmät, CVS, tms. kaatuu ja koneella olevat tiedostot tuhoutuvat
Miten havaitaan: Katsomalla tyhjää ryhmähakemistoa
Miten ratkaistaan: Pidetään varmuuskopiot kaikista tiedostoista

Riski: Ohjelmiston laatu heikko
Miten havaitaan: Suunnitelluvaiheessa tai viimeistään testauksen aikana
Miten ratkaistaan: Suunitelaan uudestaan. Jos ohjelmisto on jo testausvaiheessa, niin yritetään paikata koodia niin pitkälle kuin vain on mahdollista

Riski: Ontologioida ei saada valmiina tai yhteistyö projektiin liittyvien graduntekijöiden kanssa ei toimi
Miten havaitaan: Työmäärä kasvaa ja suunnitellu hankaloituu
Miten ratkaistaan: Tehdään ontologiat itse ja neuvotellaan asiakkaan kanssa

Riski: Valmiit työkalut, kuten DBXML, eivät toimi oikein tai odotetulla tavalla
Miten havaitaan: Työkalut eivät toimi
Miten ratkaistaan: Vaihdetaan työkalua tai muokataan vanhaa mahdollisuuksien mukaan

8 Kokoarvio

Arviot ovat erittäin alustavia. Käytetettävät teknologiat ovat ryhmälle uusia, joten vaadittavan koodimäärän arvioiminen näin alkuvaiheessa on lähes arpomista.

XML-data	2KB/kuvakortti. Kortteja noin 500 eli yhteensä 1MB
Tietokantaliittymä	1000 riviä
Ontologiat	Pitäisi saada valmiina. Ei arvioida tässä vaiheessa.
Käyttöliittymä	850 riviä
Tekoäly	500–700 riviä
Ontologioiden käsittely	700 riviä

Näistä tulee yhteensä kolmisen tuhatta riviä. Lopullinen kokoarvio tarkentuu myöhemmin projektin edetessä.