Visuaalinen tietämyksenmuodostamisohjelmisto

VETO

Projektisuunnitelma

Helsingin yliopisto
14.6.1999
Jaakko Haakana
Teemu Head
Toni Laturi
Jari-Ville Lääperi

Dokumentin versiohistoria

Versio	Tekijä	Päiväys	Muutettu
0.1	JH	1.6.1999	ensimmäinen luonnos
0.2	JH	6.6.1999	Korjattu versio, lisätty GANTT-kaavio
1.0	JH	7.6.1999	Tarkennuksia
1.1	JH	10.6.1999	Riskianalyysi, viitteitä, sanaselityksiä, GANTT-kaaviota korjattu
1.2	JH	14.6.1999	Kriittinen polku, kotisivun osoite

Sisällys

1 Tausta

Projektissa kehitetään ohjelmisto, jolla suuria määriä moniulotteista ("monimuuttujaista") dataa voidaan visualisoida joustavasti, ja joka näin tukee visuaalista tietämyksen muodostamista. Visualisointitekniikoina toteutetaan glyyfien[1] ("glyphs") käsittely sekä rinnakkaiskoordinaatit[2]-menetelmä ("parallel coordinates"). Visualisoitava data on alustavasti yksi suuri taulukko mutta on varauduttava myös siihen että data on joukko assosiaatiosääntöjä tai kattavia joukkoja.

Asiakkaana projektissa ovat Matti Gröhn (TKK) sekä Hannu Toivonen (Rolf Nevanlinna -instituutti).

2 Tavoitteet ja rajaukset

Ohjelmistossa kiinnitetään erityistä huomiota vuorovaikutteisuuteen ja käytettävyyteen käyttämällä aikaa käyttöliittymän protoiluun ja huolelliseen suunnitteluun. Ongelmana on datan suuri määrä, josta käyttäjä ei mahdollisesti heti tiedä mikä häntä voi kiinnostaa. Ohjelman on tehtävä helpoksi dataan porautuminen ja erilaisten visualisointivaihtoehtojen kokeilu. Ohjelmiston tulee myös tarjota selkeä rajapinta uusien visualisointimenetelmien ja datatyyppien lisäämiseksi myöhemmin.

Ohjelman toteutus priorisoidaan siten, että ensin toteutetaan toimiva kokonaisuus yhdellä yksinkertaisella kaksiulotteisella glyyfillä ja mikäli aikaa jää, voidaan glyyfejä lisätä.

Dokumenttien kieli on suomi. Käyttöliittymä toteutetaan englannin kieliseksi ja myös koodissa rajapintojen ja kommenttien kielenä käytetään englantia.

Asiakas hyväksyy tuotteen hyväksymällä vaatimusmääritelmän, välisuunnitelmat sekä lopullisen tuotteen.

3 Ympäristö

Ohjelmiston kohdeympäristö on Linux-käyttöjärjestelmä. Ohjelmisto toteutetaan yhdeksi ajettavaksi tiedostoksi Java ohjelmointikielellä.

[1] Glyyfi on graafinen symboli, joka ilmaisee erilaisilla ominaisuuksilla (esim. muoto tai väri) mallinnettavan parametrin arvoja.

[2] Rinnakkaiskoordinaatti-menetelmässä moniulotteista dataa visualisoidaan piirtämällä kukin ulottuvuus tai attribuutti rinnakkaisille pystyakseleille. Kunkin alkion sijainti tässä koordinaatistossa ilmaistaan piirtämällä yhtenäinen viiva alkion kunkin ulottuvuuden akseleilta toiselle.

Ohjelmisto on suunnattu sellaisten ihmisten käyttöön, jotka tietävät analysoitavasta datasta paljon, mutta eivät välttämättä data-miningista tai tietokoneella visualisoinnista ylipäänsä.

4 Organisaatio

Projektiryhmään kuuluvat Jaakko Haakana (projektipäällikkö), Teemu Head, Toni Laturi ja Jari-Ville Lääperi. Projekti- ym. kokouksissa sihteerivastuu kiertää projektiryhmän jäsenten kesken.

Ohjaajina toimivat Karri-Pekka Laakso ja Sari A. Laakso.

5 Toimintasuunnitelma

5.1 Kokousajankohdat

Projektin seurantakokouksia pidetään maanantaisin, keskiviikkoisin ja perjantaisin klo 10.15-12 TKTL:n salissa B436.

5.2 Aikataulu

Projekti tehdään soveltaen vesiputous-mallia. Poiketen vesiputous-mallista, toteutusratkaisuista tehdään huomattava määrä protoja. Aikataulu ja työvaiheet on esitetty liitteessä Gantt-kaaviona.

Seuraavassa on lueteltu oleelliset päivämäärät sekä luovutettavien dokumenttien sisältö.

Projektisuunnitelman palautus	8.6.	
Vaatimusdokumentin palautus	15.6.	
Käyttöliittymädemo asiakkaalle	16.6.	
Vaatimusdokumentin palautetilaisuus	21.6.	
Käyttöliittymäsuunnitelman palautus	24.6.	Luonnoksia käyttöliittymästä, sen ulkonäöstä ja siitä miten se toimii. Käyttötapaukset.
Käyttöliittymäsuunnitelman palautetilaisuus	30.6.	
Toteutussuunnitelman palautus	1.7.	Järjestelmän arkkitehtuuri, moduulit ja niiden väliset rajapinnat.
Toteutussuunnitelman tarkastus	5.7.	
Ohjelman demo asiakkaalle	30.7.	
Käyttöohjeen ja toteutusdokumentin	6.8.	Käytetyt toteutusmenetelmät,

palautus		testaussuunnitelma ja testaustulokset.
Loppuraportin palautus	9.8.	Arvio omasta työstä ja projektin työvaiheista. Post mortem analyysi.
Käyttöliittymän esittely	10.8.	

5.3 Työaika-arviot

Seuraavassa taulukossa on esitetty eri työvaiheiden työaika arvioituna yhdelle projektiryhmän jäsenelle.

Projektisuunnitelman teko	10 h
Käyttöliittymän ja toteutuksen suunnittelu	60 h
Toteutus (koodaus ja testaus)	100 h
Käyttöohje ja toteutusdokumentti	30 h
Projektin lopetus (loppuraportti ja demo)	15 h

5.4 Kriittinen polku

Seuraavassa esitetään projektin kriittinen polku, eli työvaiheet, jotka on valmistuttava tässä järjestyksessä. Jos edellinen työvaihe viivästyy myös jälkimmäinen viivästyy. Koska työvaiheita aloitetaan lomittain, työvaiheissa on kuitenkin hieman “löysää”, joten vaikka yksi työvaihe viivästyy, se ei vaikuta välittömästi seuraavan työvaiheen aikatauluun.

1. Vaatimusdokumentin teko
2. Suunnitteludokumenttien (käyttöliittymä & toteutussuunnitelma) teko
3. Toteutus
4. Toteutusdokumentin teko

5.5 Riskien hallinta

Seuraavassa taulukossa esitetään mahdollisia riskejä ja niiden suunniteltu ehkäisymenettely.

Riski	**Ehkäisy**
Työmäärä on arvioitu väärin ja aika loppuu kesken.	Priorisoidaan suoritettavat asiat hyvin ja aikataulujen paikkansapitävyyttä tarkkaillaan.
Tekniset ongelmat estävät projektin etenemisen	Ei jäädä selvittämään teknisiä ongelmia liian pitkäksi aikaa. Tehdään kompromisseja järjestelmän teknisen erinomaisuuden kustannuksella.

5.6 *Dokumentit*

Projektissa tuotettuja dokumentteja säilytetään projektin kotisivulla osoitteessa

http://www.cs.helsinki.fi/~veto/

6 Menetelmät ja standardit

Dokumentointivälineenä käytettään MS Word 97 –tekstinkäsittelyohjelmaa sekä MS Project 98 ohjelmaa.

Versionhallinnassa käytetään CVS-versionhallintajärjestelmää.

Java-koodauksessa sovelletaan Sunin Java-tyyliopasta *Java™ Look and Feel Design Guidelines*, joka on saatavilla osoitteesta:

http://developer.java.sun.com/developer/techDocs/hi/guidelines.html

Liitteet

Liite 1. Gantt-kaavio projektin aikataulusta ja työvaiheista

Viitteet

Sun Microsystems: *Java™ Look and Feel Design Guidelines*